A Glimpse of Heaven

Jack Edmonds

When I was eleven and twelve I was going to be a nuclear physicist, because of the Captain Marvel comic books. You know the Captain Marvel comic books? They really taught you nuclear physics, you know. All those pictures of atoms and how cyclotrons worked. My dad's girl friend, the waitress in the tavern he'd go to every day, collected comic books. And she passed them on and they taught me science.

My dad made concrete stone products. Everyone was in the construction business. My grandfather was a carpenter, my uncles were electricians and bricklayers and my dad made these concrete blocks. I had summer jobs at carrying concrete blocks around, and coping them on the tops of buildings and laying them along sides of hide-aways and so on. It pretty much blew my dad's mind that I was a bookworm.

Here is where it started. I was an amateur astronomer when I was fifteen years old and I met this wild mathematician at the Naval Observatory who said to me: "Oh, I can teach you calculus in a couple of weeks." That was Glenn Draper. And so, gee, calculus, what is that? So as a fifteen year old I would go over to the Naval Observatory and he would spend the Government's time that he was — as a mathematician — supposed to be calculating the Naval Almanac on his desk calculator, he would spend teaching me calculus. But his motive was to teach it to me his way. He didn't believe in Weierstrass and the theory of real numbers; he believed in infinitesimals. And he also didn't believe in noneuclidean geometry, he had a proof of the parallel postulate. He also later on had a proof of the four color conjecture, he had a proof of the n-body problem in physics. And he laid all this on me . . .

The Naval Observatory had this beautiful library, with a mathematics section, and he got me to reading these calculus books in the old Newtonian style, the infinitesimal way, and this nonsense with Dedekind cuts and real numbers, and he got me to reading noneuclidean geometry and so on. Well, boy, this was heady stuff for a fifteen year old!

And at just about the same time I hung out in Laudermilk's bookstore, just a dusty bookstore in the center of Washington. Right next to the Treasury, just one block from the State Department. Torn down now, what an incredibly good place.

I read everything in this store. There were little alcoves in it where you could just sit and read whatever, and I had come across this book on Euclidean geometry and proving stuff. God, you know, to prove something, the whole thing blew my mind and . . . Hell, I have been kept in at recess all my life for not being able to do my arithmetic. But this idea of proving was really thrilling.

Not only was the idea of proving stuff thrilling, but as soon as I am beginning to absorb, this Glenn Draper comes along and teaches me that mathematics is a serious, philosophical, controversial thing, and that most mathematicians are jerks. And so my head is spun with his proof of the parallel postulate and all the things he was trying to teach me.

He had written a book called *What is Truth?* I still have it. Later on he got fired. I guess that he got fired for thinking more about things like this than calculating his Naval Almanac. I went to see him, I guess it was about six years later when I was visiting, back from university. I visited him and he had been fired and he was pretty down. So mathematics for me in this way became a matter of the underdog against the system.

I also belonged to a science fiction club, with a lot of weird people in it that really believed in crazy ideas about living and the earth. And another big thing, dianetics started then. L. Ron Hubbard wrote this book, that was really a breakthrough. Then the other big thing besides the living and earth and dianetics, was the world of nonaristotelian logic. That really had glamour. So that stuck with me along with Glenn Draper's thing.

* * *

I got a really good scholarship to a classy university, Duke. But the math courses were just boring, working the way backwards. We were doing trigonometry and freshman calculus again and I didn't like the math. I took humanities courses, I took everything. If I had the abilities I would be a writer instead of a mathematician. I am a mathematician because I am too slow to be a writer. But I spent one summer writing a play for the University that was rejected for a musical. You know how universities have these musical comedies. I wrote *Hiho Sputter*, it was a kind of a takeoff on the Lone Ranger, years ago. I still think it was a good idea for a musical but it wasn't accepted. And I directed a play at the University and wrote long treatises on philosophy.

I spent as much time in philosophical discussions and dilettante this and that. I got in battles with a professor in political science. He had just returned from Austria or something and I solicited an article from him for the students' magazine. I felt like I had to reject this article.

Anyway, many really exciting things like that, and the marks didn't matter a bit. I lost the scholarship by the third year, and yet I really liked studying. I dropped out many times in my career and this was the first time.

I had it worked out that I needed to be a gentleman scholar. Part of my philosophy was that we really ought to be able to live in a university environment all our lives. That is the way life should be. I would go out and I'd make a lot of money. And then I'd just come back and be a gentleman scholar after I was rich.

So I invented some things, several toys, and I went to New York City. This was as far as I had ever traveled away from home, the trip from Washington DC up to New York City. And I went around trying to peddle my ideas, and one company took me to lunch and bought me a martini and I signed a contract. And a few months later I saw one of my toys for sale there, with their name on it, but it turned out that it was really a different company now. They had gone bankrupt, and so my contract didn't count.

Another of my toys was a string of triangles. These triangles were hinged around a row and the middle ones have slots or something, so that the outside ones can attach into them. We can make pictures of these things. This toy is the beginning of matching theory for me and so, maybe, the beginning of my interest in algorithms.

You see, these ends will attach in here, okay? And so you have a whole bunch of these pieces, and you can put them together, and flip it over and put them together. So, then there is the question: Can you make any polyhedron of equilateral triangles from these pieces?

You see, after you've pieced them together, then since these go into these, what you've left is each of these hinges, a pair of triangles. And these pairs of triangles are hooked together and what you have here is: this pair of triangles has a flap on it like that, and they had to be pieced together so that the flap sides go to the nonflap sides. And so: Can you build anything out of these? Well, you can't, but can you build any triangulation which is combinatorially a planar map?

If you can draw this map of triangles in the plane, then can you realize it by putting these pieces together? It is quite a puzzle to try to get them together to do that. Well, you see, the first step then is: Do you have a matching in the dual graph? It amounts to replacing each of these triangles by a point, and making the dual graph.

So this is a good toy. Well, when you design such a toy, the first question is: You've got this three-degree graph here, the planar dual of this; does there exist a perfect matching in that graph? That is the first part. Then after you have the perfect matching, which gives you a pairing of the triangles, then you have to worry about: Will the flaps go right so that flaps on one go into the sides of the other?

Okay, so that was my first contact with matchings and I have screwed around with this and I found . . . I would like to be able to say how smart I am that I proved that every graph with three edges meeting each node and no cut edge contains a perfect matching, but I didn't prove Petersen's theorem.

I needed this and somehow or another I came across a pop article on graph theory, and that is what brought me to this toy. And what I had to do to prove that the thing could be built, was to use Petersen's theorem together with an argument about how the flaps fit together. So it was this article, which I probably had found in one of those books that get you interested in math. You know, I was reading books, like I say I was hanging out, and while you were on the basketball court I was in this dusty bookstore, Laudermilk's, you know.

This toy is a darned good application of graph theory. It still hasn't been marketed, maybe somebody will. I don't care anymore about making enough money to be a gentleman scholar, so anybody can make this if they want.

This is what I was doing, this is what I was doing, damn it. To make toys to sell to these toy companies to get rich so I could go back and do university the way I wanted. I was going to spend the rest of my life just being a student, I liked the student life. I was going to pay for it and that's what I was doing this for.

But I didn't get rich during that year — of course dad's concrete blocks were still supporting me while I was getting rich. Which wasn't a lot of money. And various other ideas flopping, I never got rich. So I never was able to become a wealthy scholar.

It was during that time that mathematics became fun again, just playing around. When I was at university I had left mathematics because it was just so much more fun to direct a play and this and that. But mathematics is fun if you don't let mathematicians push you around when you are doing it. So I was studying it on my own during that year.

How can a person resist picking up a book called *Metamathematics*, this is the theology of mathematics. So I started into Kleene's book by this name and had a heck of a time trying to read it. It really got my imagination going about the foundations. Well, I guess Glenn Draper years before had caused me to worry all the time about philosophical questions about mathematics, as well as these dianetics people who believed in diros and nonaristotelian logic. So I took up metamathematics during this year off.

I went back to university then for my final year, my marks were good enough to transfer, I hadn't lost anything. I went to George Washington University. I signed up for a course in French literature and French novels and translation, and I got a job as a so-called copyboy, errand boy, gopher, for the newsroom of the *Washington Post.* I really wanted to be a writer if I wasn't so damned slow in writing. So I was a copyboy late at night and took these courses. But two things happened.

First, I couldn't get through all these novels. To pass, you had to read seventeen novels for this one course, and write these long essays on them. And I just couldn't do it.

And another thing that happened was that as a copyboy at the Post I was with all of these humanities type kids, you know, from Stanford and other classy places, who were trying to break into the business. And one of the women, and I fell for . . . , well, this group of people, what impressed them was when I told them I was a mathematician. And you know, a would-be-writer/copyboy, that was everybody in the crowd. But, I mean, here is Einstein, I was their oddity.

So I guess I saw the light there and paid more attention to the math side and decided: I might as well give up on my literary ambitions and try mathematics again. You know, I liked it a lot studying it on my own.

* * *

So that got us to the graduate school. It's a darn good thing, University of Maryland was there for me, and I started studying analysis formally. The program at the University of Maryland was pretty much analysis and applied mathematics in the traditional sense. But still the individual contacts like with Marcel Riesz — he would talk about anything, and so on.

There was a supposed research seminar there. People would study papers and report on them, whatever. And I remember I thought that I had my own theorem there, so I gave a talk on my own theorem in abstract algebra. And then after this seminar I realized it was false. But I wouldn't dare to say anything, and nobody caught me. So that was a good introduction. You know, that has happened to me since.

My Master's thesis was in graph theory. I had been reading Coxeter and Moser's book on generators and relations for finite groups, I guess because it had nothing to do with these analysis courses I was taking. The big idea there is to construct surface maps with a lot of regularities, to represent a group that is given by particular generators and relations. Well, that is neat stuff. But, incidentally, in there, Coxeter and Moser mention that any finite graph can be embedded in a closed surface so as to divide it up into disk-like regions, and gave a reference to I guess was a student of Coxeter's. I didn't go to the *Math Monthly* to look up this reference until later. I started thinking about this.

Why could any graph be embedded in a surface? I got the idea that, well, you could put at each node any cyclic ordering on the edge ends at that node and that determines an embedding. You just walk around turning the corners, and let the walks close up where they may, and let those walks bound disks, and you get yourself a surface.

It was really hard for me to visualize. I mean, I wasn't at all sure that it was really true until I got this idea that we can draw a picture allowing edges to cross and order the nodes any way we want to, and then we can do this with a pencil. And it wasn't until I got the idea of that analogue algorithm for doing it, that I realized it's got to be true, and I translated that into combinatorics. So that was my Master's thesis.

This was nice, and Professor Reinhart said: "That is nice. That makes a good Master's thesis", and I appreciated that. But I sent this away to Coxeter. I wanted to study with the great geometer Coxeter. He had written this book called *Polytopes* — what a name, huh? — and I had looked up that paper by his student that he cited. It proved that every graph can be embedded in an oriented surface. And there the proof is: You delete enough edges till you have a tree. Of course, you can lay the tree on the surface and now you start laying down all the edges until you get to the point where you cannot lay down an edge without crossing an edge that is already there on the surface. Then you build a handle on the surface and lay the next edge along the han- . . .

Well, that is a stupid proof, right? So I sent this off to Coxeter and I said I wanted to study with him. And he wrote back that he didn't believe the theorem and he sent me back a counterexample, and that really made me proud.

But anyway, I didn't get to go there. He said he didn't have any funding for me to study in Toronto. I don't guess he ever believed the theorem, but he believed an application I did of it to a group conjecture in his book.

I was able to construct a counterexample to his conjecture about regular surfaces by just taking my theorem about putting cyclic orderings on the nodes. I got a systematic way to put a cyclic ordering on a graph with seven nodes. I could just use a field on seven elements to construct a symmetric ordering of all these things and I got a symmetric map like he wanted.

When he came out with his book called *Introduction to Geometry*, he included my symmetric map, but he never got my theorem about how you get maps. I mean, if I were to write a book on geometry, that is the part that I would put in rather than this particular map, but anyway.

That was my first taste of mathematical fame, in that book *Introduction to Geometry*. So that was nice.

I left graduate school after I drifted away, as at the same time most of my classmates did. The philosophy of the PhD program was that they needed bargain teachers and maybe some of the students would make it. It was a large state university that needs lots of teachers, University of Maryland. And this is a big chip on my shoulder, the idea of a PhD program that operates on a basis of lots of students and maybe a few of them will make it. In the meantime they teach and so on.

I had two advisors, they were good guys. I had Bruce Reinhart who got me interested in topology. And I had Marcel Riesz who was a grand old man of functional analysis. I wasn't smart enough to do the things that he is most famous for, but he was pretty old and I was his research assistant and my job was to take him to taverns. He was alone and about seventy years old in College Park and my job as research assistant was to care for his transportation needs and whatever. And in the meantime it was really a good experience for me, doing geometric problems. It was fun to hang out with him. The only thing that stopped that is that I got pregnant and he and my wife didn't like each other. You know, they were kind of competing for my attention. That was '60.

I liked my professors there a lot. I especially liked Horváth because he would wave his arms around, with a lot of enthusiasm, and not giving a lot of details that I had to write down. The professors there were really good. Who knows what bad thinking went into the design of the program, in such an environment of hundreds of students on their GI loans. They needed these many, many PhD students. On the other hand, I would never have been a PhD student probably, except for such a thing.

There were some courses like measure theory. I flunked the course because I thought that the Halmos development of measure theory is just stupid. People like Halmos so much, but it is just such pedantic abstract crap, you know, with just abstraction for the sake of abstraction.

I mean, my feeling is known for that. Measure theory was a required course and I flunked that one, and then I flunked another graduate course, from my favorite teacher Horváth, and it was because — I don't survive on ability, I survive on taste. And Horváth, I liked his lectures so much because there was all his arm waving, but I wasn't able to translate his really great

ideas into the answers for the exam and I decided: I am sorry, I don't understand this sufficiently, I would rather take the course again. So I flunked that course too, and this was complex analysis, a great subject. And I took it a couple of more times, from different people and different versions of complex analysis. I really liked it, but, anyway, there is the gentleman student again.

* * *

So I started having children and as I say, my wife didn't get along with these mathematical people and particularly the grand old man so much. And 1800 dollars a year really wasn't enough to support a family. So I got an apartment in downtown Washington for sixty dollars a month, and I found this job at the Bureau of Standards, and that was great. That is where grad school really started for me, with Alan Goldman.

You know, if I hadn't just by luck got this job with Alan Goldman, I would have been a dropout like all my classmates. I've never heard from them again.

There was a mathematics division, and it was a wonderful thing. Alan wanted to start this operations research section of the math division, and I was able to join him for that. You see, the Bureau of Standards got research contracts from other government agencies like the university would. And it was my job to figure out a way to assign radio frequencies to airplanes or something, so that they don't interfere with each other. So I had a packing problem there. And that got me started on thinking about combinatorial optimization.

I ran around whatever I could find on combinatorial optimization, and there was Gomory's cutting plane method. It just "solved" the ILP problem, you know. And then a guy named Roth at IBM had this long elaborate paper for designing switching systems, where you think of the different states of the switching system. I mean, it is Boolean formula simplification in terms of the way they teach switching circuit theory in books, and yet he was packing all these hypercubes in cubes, by an enumerative thing. It was really elaborate and I wasn't impressed by this. I really wasn't impressed by this and I really wanted to solve the integer LP problem. As I say, there were cutting planes.

It's too bad, you know, Gomory had a beautiful thing there. In this is a neat intellectual idea. Now, if Gomory had presented the following: For any polytope — for convenience assume the polyhedron is included in the nonnegative orthant — you can get the integer hull of it by taking a valid inequality, and pushing it until that inequality somewhere bumps into an integer, and then taking more valid inequalities, which, by Farkas, you know are a combination of the inequalities you've got. If he had stated as a theorem that the integer hull of this polytope could be obtained by this process ... Well, that idea became later really inspiring to combinatorial optimizers. But his theorem was: Here is a finite algorithm for integer programming. Well, I mean, a finite algorithm for integer programming? That is not interesting! That is not interesting!

And Gomory knew that integer hull theorem! I heard him lecture, I heard him say it! No question that he said that! No question that he knew that! He preached that, this is what he taught me! But this isn't what went down when T. C. was writing his book on integer programming, say. I mean, there are no theorems in that book, huh?

I came across Berge's and Norman-Rabin's augmenting path theorem as part of my search for anything to do with combinatorial optimization. And I generalized it to an alternating tree theorem for set covering. It's a nice theorem. It immediately includes the Berge-Norman-Rabin theorem. It is a tree where degrees of the appropriate nodes are of degree two. And so, when you interchange the outer nodes with the inner nodes you automatically get a smaller set covering, huh? The theorem is, if this set covering isn't smallest, then there exists a tree of this structure.

Balinski wrote this survey on integer programming including these theorems of mine. I was very proud that he was presenting my work, consciously. In there, there was also a node packing version of it. I was never conscious, I was very proud actually. This was very nice for me, when I saw this survey on integer programming and he had a section on my theorem in there. It is definite, no question of independence in this rare instance.

Ray-Chaudhuri at IBM had another generalization of Berge's alternating paths, it had something to do with alternating sets. He used paths but they weren't really paths, I never really understood it. It must have been something like what I was doing but I didn't understand his version of it.

But we're missing the point here, a really fundamental point. I am proud of taking this philosophical position here. Ray-Chaudhuri was giving it as an algorithm, at least in my recollection, and Berge was talking about his path theorem as an algorithm, and I knew, you know, I knew my alternating tree theorem wasn't an algorithm, and I liked that. As far as I was concerned Gomory's cutting linear combination sequence, I mean, it's an algorithm, but ... I wanted these standards of what isn't obvious, as opposed to what it means, an algorithm for a problem that is obviously finite. And here are all these OR people, electrical engineers or whatever, they were writing all of these, try this, try that, you know. So I was very proud that here I had this idea for what could have been presented as an algorithm but that I wasn't going to present as an algorithm.

Now the first time I ran into Dick Karp was when I was talking about this

theorem at the Math Society meeting in New York, and he knew Ray-Chaudhuri and he knew this guy Roth that had this Boolean circuit elaborate enumerative theory for simplifying Boolean circuits. I was laying on Dick these ideas, to say you're having an algorithm here for a finite problem isn't saying anything. We got to look and find something interesting to say. And one interesting thing is: Is there a polytime algorithm, and that is when I started waving this banner about ... Kind of like Gene Woolsey later was ... He got up at one of those Math Programming meetings and made fun of all of these things like the cutting plane method. The cutting plane method needed that.

How I came to these ideas? I don't see how anybody could miss it! I mean, nobody could miss that! The main point is that mathematicians don't present ideas, they only present technical achievements. That is one thing. And that is a damned shame.

And then another crucial aspect of it is: The reactions I would get when I was ranting about this at that time — I remember my obsessions and my talking at full tilt —, the biggest reaction I got is: "Well, it's kind of silly to expect such a thing, and let us see, it doesn't have any real meaning, oh, and so what, if it were n to the 28th, you know, that doesn't...", and all that kind of stuff. I just collected these kinds of reactions that weren't questioning it as a natural phenomenon. Oh, and another response was, I think it was Ralph's response: "Well, you don't have such an algorithm for linear programming!" I think really crucial to the success of the cutting plane algorithm was the just phenomenally unexplainable success of the simplex method. And with that tradition of that success, people weren't ready to ..., they were talking about linear programs that automatically have integer solutions. The criterion for niceness was that a problem was automatically a linear program.

I know Munkres talked about the polynomial efficiency of his transportation method, which was nice. I don't recall Dijkstra's paper but there is no question that he was aware of polynomial efficiency, and certainly Munkres did. Do you think Munkres paper gets what it deserves? I think he was an outsider, right?

Well, anyway, there was the challenge of this ILP problem. Everybody — I don't know about everybody, but it seemed to me like everybody — was praising the cutting plane method as having solved the integer programming problem. And the context of this was, you know, it isn't trivial, it isn't at all trivial, it is really heavy math in there, it is great mathematics, and it is just using the simplex method! Well, I cannot say why it was such a smash but anyway, you remember people just fell hook, line and sinker for it.

This is what I had never heard. There is the polynomial — I think I said algebraic — time algorithm for the optimum assignment problem. Is there or is there not a polynomial-time algorithm for obviously finite bounded integer programming? I mean, this is my sermon: Is there or is there not? And it is one thing to have an algorithm and announce bounds on it. I don't know why it's a deal to discuss 'or is there not', the idea of proving if it is not, but it took some doing to sell Knuth on this, and other people. Isn't there an algorithm?

I mean, this is what we've seen: Dantzig has shown us the idea of reducing all these problems to a central fundamental problem. And we can reduce it down to a more particular problem if we want to. I mean Dantzig was really, really important here, I think, and this idea of reducing was continued on to the sixties by the OR community. But like the traveling salesman problem isn't at all obviously an ILP. And even setting it up as such, this integer programming isn't solved.

Gomory's beautiful mathematics, what he is saying is not that he has solved integer programming. And that is an important question: Is there a good algorithm or is there not? Anyway, that was the sermon.

And then the next part of the sermon — and that was really important to me — was this certificate as opposed to an algorithm. Now I don't think certificates are so blatantly there in most areas of mathematics. But they are in linear programming. I mean, it is hitting you right in the face.

This marriage theorem here, god, I spent hours to understand it, reading, maybe it was in Ryser's book or a book by Marshall Hall. For any, any ... I forgot if they say it in terms of a graph or a zero-one matrix, ... oh, it was a system of distinct representatives. That is it. Here you're given a family of sets. Then either there is a subfamily which is bigger than its union, or else, you can pick one element out of each set, all distinct. Is that interesting? Is he saying something there? I have spent hours to see: Is there any content to this?

Now, the impression I've gotten from the way a lot of theorists might think, is in terms of: Is this obvious? or: Can I prove it? But I couldn't decide whether it was obvious or not. I had absorbed it by thinking of it as: Well, is it the idea that it is a good certificate for not being able to pull distinct elements of those sets? Now here, you know, I've to assign these frequencies to radio stations and whatever, and I cannot do it, and, I mean, this theorem is saying that either you ...

No, what is wrong with that theorem, is that usually it isn't stated the way I just stated it. This is what sorted out the way it is always stated: You can pull a distinct element from every set, if and only if for every subfamily, for *every* subfamily, the cardinality of the union is at least as big as the subfamily. For *every* subfamily. Well, now, gee, right, is it easy to pull a distinct element from

each set? Is it easy to tell when you can pull a distinct element from each set? Is it easy to look at *every* bloody subfamily, you know, and, I mean, that is the way I think most people look at that theorem: if and only if for *every* subfamily. And the light hit me! This is not about *every* subfamily having this property! This theorem is telling me, either you can pull a distinct element from each set, or, *or*, there exists *one* subfamily that makes it easy for me to see that you cannot. And, boy, I mean, you know, this certificate of this one subfamily. So that was the only way I could appreciate that theorem, before whether it is easy or hard to prove.

And so, Farkas' lemma, *NP*, and so on ... This is what we got to do with integer ILP's, isn't it, we need a certificate. So here is this x. Is it optimum? Does it maximize? Does it or doesn't it, you know. Well, if it doesn't it is easy enough to see that it doesn't. If it does we need a certificate for that.

Now look, here is the amazing thing. Hilbert, you know — I learned this stuff from my Marvel comic books and my Laudermilk's bookstore — Hilbert in 1900 asks: Is there an algorithm for such and such a problem? He stated explicitly in various places that he firmly believed that a well-posed mathematical problem had to have an answer. But here is the amazing thing. He was the greatest mathematician. We really see this in lots of ways. He was asking this question, is there an algorithm for this? But he didn't know what an algorithm is! And it took all these smart ..., Gödel didn't know what an algorithm is, Church didn't know what an algorithm is. It took thirty-five years until ... No, it took ..., Euclid, I mean, Euclid presented the Euclidean algorithm. Cardano or someone presented the algorithm for solving a cubic equation. Mankind had been presenting algorithms for thousands of years, the Arabs invented the word. And all of this stuff, all of those analyses and stuff being formalized in, you know, trying to make calculus believable by these ridiculous Dedekind cuts, and here they are coming up and they don't know what an algorithm is. Now that is what is incredible, that is really what is incredible! All that time, thousands of years, the mathematicians with their algorithms and Hilbert was saying that we want an algorithm that ...

Alan Turing invented the computer. He literally invented the computer. That is what a Turing machine is. Turing submitted this incredible paper in order to get a postdoc at Princeton, as well as his thesis on some abstruse group theory. And it was von Neumann's job to read these things to decide if he should be a postdoc. Now von Neumann said: "Yeah, yeah, he should be a postdoc. He wrote this good paper on group theory." Von Neumann didn't mention that paper, didn't mention that paper that invented the computer and proved that there is no algorithm for something. And to many people von Neumann is known as the father of the computer, for the reason that he started building Turing machines. I have heard that later on in conversation he would mention the early influence. But I mean it really takes a while for it to sink in the incredible thing that Turing did. It is important that operations researchers realize that this guy Turing ... You've got to read this book by Hodges about him. It is just an incredible story.

And this Von Neumann Award was ... You remember how turned on I was about it, and how gushy I got about it. It was the biggest thing that ever happened to me. You know, I don't mind flattery but I don't like von Neumann, he is not my kind of guy. Head of the Atomic Energy Commission ... Anyway, there are a lot of people I don't like and it isn't my business to lay out my overabundance of dislikes. But it reaffirms that what we are talking about here is an underachiever who had a chip on his shoulder.

These Turing machines were suddenly all around and they were being built and I am just a child. I am just a child that sees them there. I had nothing to do with them. I am just a child that sees these computers around that same year that I dropped out of college. I tried to get a job at IBM and they didn't accept me and I continued on with this real antagonism. IBM has been my nemesis ever since. I took this intelligence test and I like to think that the reason that they didn't hire me was that during this year I dropped out I was not working as an undergraduate. I like to think it was because back then ... They had these signs that said 'Think', they were all wearing neckties, and they were real ickysquares and I, I was a beatnik. I like to think that that is why they didn't hire me. But they probably did not hire me because I didn't do well on that exam where they gave you six numbers and you had to choose what the seventh number was and that stuff.

In retrospect I say Turing did this for us, after two thousand years. I am beating, I am waving these banners. I am talking like crazy about this point. In retrospect, I think the moral here is that mathematicians have got to defend what they know in terms of technicalities. A lot of people concerned with algorithms don't like threats, whether it is from Gene Woolsey or from me. On the other hand: Why, why do we ask the question: Does there or does there not? I mean, it is obvious that it is a precise mathematical question. But why was there no response to that question? Because people don't present stuff except for technical accomplishments, you know. You don't get brownie points except for technical accomplishments in this game. Mathematicians don't like philosophers.

* * *

You know, the luck of my life ... I take it back, the luck of my life, because I have already spoken of a half a dozen lucks of my life. That crazy philosopher that taught me calculus as a fifteen year old, and just drifting away from a legitimate grad program to Alan Goldman. But the next luck of my life, and this was really crucial, was because of this alternating tree theorem. Alan sent it to Tucker, and Tucker sent it to Fulkerson, and Fulkerson was organizing a summer workshop at Rand Corporation on combinatorics, summer '63.

So Alan set this up, I am off to Rand Corporation, except that the Bureau of Standards does not give me leave. And so here is another thing I am really proud of, you know: No way I am going to miss being with this workshop that Alan set up for me! Alan was my immediate supervisor, he was my total supervisor at NBS and set this up for me, no way I was going to miss it, and so I quit the Bureau of Standards. I quit my four thousand dollar job or whatever, and immediately applied for a new job at the Bureau of Standards, and went off to this workshop. I was really pleased, by the time the workshop was over I was hired at two grades higher. And I was making six thousand or something like that. And that is nice.

But this workshop, oh geez, there were about forty or fifty senior mathematicians there, and they were everyone in combinatorics, the senior older people. Oh gee, that was so wonderful ... The only four young people were Larry Brown, you know him, and Balinski and Chris Witzgall and me. We shared an office there. Now Alan Goldman had reached out to these colleagues of his, and recommended me and he was guiding me along. So I got in discussions with Dantzig and he, more than anyone at the workshop, took an interest in arguing with me.

I had this kick about I didn't have an algorithm for set covering, I had an interesting theorem. What I needed was a certification, alright? Well, there was a certificate for whether a graph contains a perfect matching: Tutte's theorem. That beautiful generalization of this theorem of Petersen, that enabled me to make my toy. And there is a certificate for you. And I needed a certificate for when there was not a partition into a given family of sets or a certificate for a set covering of a given size. I needed that certificate and I needed an algorithm for matchings, and Berge presented this alternating path theorem as an algorithm for getting a largest matching.

Now this is technically important. It is really hard for anyone to see that it isn't easy that when you've got a matching in a graph and you are starting at a deficient node, that you cannot just grow a tree looking for a Berge augmenting path. I don't know if he's ever gone through this. The hardest part of that is to see how the obvious algorithms do not work there. And in later years I have worked out the various obvious methods — when I lecture — and why they don't work. But I don't know if that has ever gotten across to

Claude even. Maybe Ralph Gomory was saying: "Well, polytime algorithm for integer programming. Well, you have to get a polytime algorithm for linear programming, you know." Well, I was saying: "An algorithm for alternating trees, Berge's theorem isn't an algorithm either, you know, it doesn't give us what we want." My theorem was just as bad as Berge's theorem. Neither one of them gave us a good algorithm.

And that got me to thinking about the technicalities when we were talking about this at Rand. Talking about it with George Dantzig . . . , I so much wanted to get an algorithm. I wanted to get an algorithm for set covering and I felt sure, this theorem was going to give it to me. I had to do it with the path theorem, I had to do it with Berge's theorem. And I am sitting there in the office at this workshop. And, you know, the wife and the kid out on Muscle Beach there. I remember Balinski at this desk behind me and Chris over here and Larry there. I was drawing these paths and . . . , oh, oh, the next day, the next day I was due to give a lecture on my alternating trees for set covers. People at this workshop took turns to give a lecture, of course. So here I am sitting, and so, help me, help me, . . .

God damn it! You shrink, you know, you shrink! And . . . and . . . I had this good algorithm. I had this polytime algorithm, you know, I had the technicalities I needed. I had the small technicality I needed, as this weapon. And so I gave this — God, what a high — this lecture.

Now I like to remember I said "Eureka". I don't know what I said there, but it was like that with these guys around. At this lecture, everybody I'd ever heard of was there. Gomory and Dantzig and Tutte and Fulkerson and Hoffman and Bruck, he was the heckler of me during the talk actually. And I gave this grand philosophical speech here: Well, here is this augmenting path theorem, a special case of this augmenting tree theorem for set covering. It is an interesting theorem, you know, it is an interesting theorem. *But here is a good algorithm, here is a solved integer program.* And, you know, this was a sermon, this was a real sermon. *Here is a solved integer program.* It was my first glimpse of heaven.

The reaction was very respectful and positive. Bruck, whose thing was not algorithms but design theory you know, he was a jolly old guy. He heckled what the hell I was talking about. But the algorithms people really were supportive afterwards, they were encouraging me in this grand mission. And so that was really nice. I knew by this time that LP duality was a certificate for optimality and I had all these collections of certificates, you know. And in a situation like that, how could you not believe that there is a certificate for ILP feasibility? And I still haven't, I still haven't accepted it . . .

The easy part then was putting weights on the edges, combining this with weights and getting a certificate for optimum weight matching, which is much more impressive. I had this largest matching is equal to this minimum covering of these sets which each had an obvious upper bound, and to translate that into an exponential collection of blossom inequalities, I mean, that was no sweat at all.

I didn't know anything about the simplex method. I didn't know the Ford-Fulkerson algorithms. I learned probably that Fulkerson was writing about zero-one matrices. You know, he was big on combinatorial applications. I did know the LP duality theorem as a certificate, but I didn't know the simplex method. I heard lectures on it, but I am a slow learner. I was kind of faking it among these guys, you know, trying to hide the fact that I didn't know this stuff that they all knew. But I remember Alan Hoffman saying: "You know, linear programming, it is a really good subject. It is good theory, you will like it, you know, just sit down and master those basic . . . " We were having coffee one morning and he said: "Learn linear programming, learn it well."

If it is the case that for every objective function your algorithm will be certified to stop by this kind of this duality certificate, then this is a proof of the theorem that every vertex of the inequalities is one of the objects you are optimizing over. And I think that is nice, you know. I think the accepted way in the context of Hoffman's work on using LP's to prove combinatorial theorems is: You prove the vertices and then apply the LP duality theorem to it and you get a min-max. But this idea of reversing it, you know, you've got your certificate. It just implies: If for every objective function you can prove that one of these objects is optimizing, you've got this proof of a polytope.

It was not that big a deal to generalize the certificate to *b*-matchings and then generalize the algorithm. But the right-hand sides really bothered me there. You know the old story about Pythagoras hiding the square root of two? I wanted, I definitely wanted to hide for the OR seniors that I didn't know how to take care of those demands in polytime of the number of characters. Because I wanted to sell, I had gotten total encouragement from these OR granddads at this meeting and I was trying to show off to these IBM'ers and these computing theorists, and they did not respond.

* * *

Al Tucker next said: "Why don't you come be a research associate at Princeton next year?" God, you know, I mean, I am a two-time university dropout. But that was his response to this lecture and I did that. The next year I had this job and it was okay, the Bureau of Standards let you go off to an academic post. So not only did I get this good new job, back to the Bureau of Standards immediately after shrinking those blossoms, but I went to

Princeton! In the meantime I wrote up all these things, I worked out this weighted version and wrote that up.

Dantzig called me up and said: "Hey, I've got this bright student here who really digs matchings, and can he come work with you?" So Dantzig was sending me Ellis Johnson. And that, you know, that was really great, that was really great.

Here I am a research associate at Princeton and I am going to lectures on topology by Milnor and Fox, and the neatest thing is, Tucker says: "Alright, okay, you take care of the games and combinatorics seminar." Now I had heard of this, this was a tradition at Princeton by that time, of course the emphasis was games I guess, but Tucker had made it games and combinatorics. He said: "Alright, you take care of that."

And so I started inviting speakers from contacts I'd made at the Bureau and at this workshop, and people were glad to come. I started inviting everybody that I would want to talk, to come down. People generally were willing to come to Princeton to give a lecture, even though I didn't have any funds for this. But what I discovered right away is that there are lots of little seminars going on at Princeton and nobody has time to go to them. And so these really good people were coming and yet there was no audience for this seminar, and so I was really in a bind here. But I am pretty pleased with organizing this very early combinatorics meeting. I had Victor Klee and a whole bunch of other people invited to this, Morris Newman, he is a good mathematician. He was sort of the guy we looked up to at the Bureau of Standards, including Alan did. He later became a math professor in California some place. But I invited him up there and he had a good talk and no one came to it.

The way I got out of this, I sent out letters, okay, plan B: "We are going to have this meeting in the spring. And couldn't you come at this time?" So all these guys came and the idea was, well, you know, listen to each other, huh? There would be an audience for each other.

And we had a fantastic combinatorics meeting that spring at Princeton. I met Alfred Lehman there, I don't know where he had come from but he appeared out of the woodwork at this combinatorics meeting at Princeton. That was really a crowd of good people there. That is probably my first and last success as an organizer, you know. But I just wanted you all to know. No, it wasn't my last time. I did it again once. I just want your guys to know this, you know, I wasn't always the...

Tutte had lectured on matroids at this Rand workshop. His lecture on the dual of a graph that isn't planar, that is a very important aspect of matroids. But it was really Alfred Lehman that in his work made me discover for myself matroids as an algorithmic tool and matroid optimization. He had the switching game and that is just so beautiful.

Thinking about this Shannon game, what makes it go in terms of this idea of certificates? I mean, I have had my big first success now. You postulate a good certificate, you take a problem that is obviously *NP* — in modern terms — and you discover an alleged *NP*-predicate for the negation. And now you get together an algorithm that finds one or the other, right? And I am going to do this for integer programming. But in the meantime you know there is all these other things to do it for. Nobody else is doing this and the world is just full of it, must be just overflowing with things I can do this to. You know, we found the theorem algorithm machine now, right? Right. And the thing is, Lehman's switching game — he called it Shannon's game — he replaced it by this matroid that is just an oracle. And something made this game work. The definition of a matroid makes sense only if you think of it in terms of: You've got this oracle presenting the matroid when you ask it the right question.

Alan Goldman had asked me if I could describe the convex hull of the spanning trees of a graph. Now, geez, you know, half of my mathematical career starts there, with the matroids thing. I had done the matching polytope thing. And he asked the question and I immediately answered it and then I was ready for matroids, right? And I did it for matroids. What just gave the greedy algorithm for matroids? A certification for it is going to be a system of inequalities, just as the matching, and it is so much simpler, you know.

But then I was thinking: What makes Lehman's switching game work? That was heavy, and getting algorithms and certificates for the matroid partitioning associated with that. Boy, that is really a new ride, with oracles and so on.

I immediately started organizing a workshop — I am really into workshops and meetings and this kind of things here now, you know, the Rand workshop and Princeton and the wonderful combinatorics meeting at Princeton that spring and meeting Al Lehman. And we were going to have a big workshop at the Bureau on matroids. I think it was January '64, or somewhere in there, it must have been '64, because my own papers appeared in '65. Alan went for it, so I started looking around everywhere I could, to find anybody that may have something to do with matroids or might be interested in matroids if they didn't.

And, you know, I really give Ray Fulkerson a lot of credit for that. He was the big shot that came to this workshop. Tutte came, he was the honored main speaker and I had persuaded him to come. His papers in the *Transactions* were totally unreadable to me and I had gotten the feeling that nobody else could read them either. I wanted him to come and to rework this stuff. So he did that. He did a beautiful job of his lectures on matroids.

And I was all set with the greedy algorithm for optimal weighted independent sets. And, I mean, all these years, operations researchers — or of course, all these years, it probably wouldn't have been long at all, but since it was before I was born, it was all these years — talking about minimum cost spanning trees, it is the same thing as minimum cost bases of a matrix! Do math programmers care about bases in matrices or don't they? The same algorithm works for any weighting of the columns of any matrix over any field, it doesn't matter. And then immediately, it is no problem to get that polytope, since that is the way I am thinking of these things, from the matchings. But then the heavy stuff was this matroid partitioning which I worked on. So I had my thing to contribute, about matroid algorithms.

And Lehman was lined up, and Minty, and I guess too Henri Crapo, who was a recent PhD student. And so we had our matroid workshop, and a really good group of people came. Neil Robertson, he was a grad student at that time and he was a wonderful scholar for many years before he was recognized for it. And there were a lot of people there like that. So that was a big thing.

I had met Chris Witzgall at the workshop and through that I had succeeded in getting him to work with me at the Bureau of Standards and that was really great. I really wanted Alfred Lehman to work with me at the Bureau of Standards too. You know, me and him and Chris, sharing an office, that would have been great.

But there had been a tradition at the Bureau of Standards — I don't know where it came from, something about communism at the Bureau of Standards, I don't know, but anyway — there is some tradition at the Bureau of Standards about taking an oath of office. Al reported for the first day, and the bigwigs asked him to take this oath of office, which is nothing more than what you do in school, in elementary school. You know, saying that you pledge allegiance to be loyal to the US. But Al wouldn't do it. Alfred wouldn't do it. And Ida Rhodes, oh, what a woman, she was the grand dame, the grand old lady at the Bureau of Standards for computers and using computers for automatic translation. At that time, machine translation was perceived as a really main use of computers. Anyway, Ida Rhodes, the elder statesman of the Bureau of Standards was there and she started tug- . . . , this is funny, she started tugging on Alfred's arm. She literally grasped his arm and said: "Oh, come on!" And Al, he got his back up and he refused to do that, and I guess the Bureau got its back up, and didn't hire him. And I mean, he was hired. And he made a big noise about the Bureau of Double Standards . . . And he took it out on this matroid meeting.

I was proud to write a paper on matroid partitioning and Lehman's switching game, you know. He might call it Shannon's switching game, but to me it was Lehman's switching game. To me he is the hero of the subject.

But he had a lot of pretty serious views on things besides mathematics. He has hurt himself over the years. You think I hurt myself with my hang-ups, but he . . .

After that, he got a job at Walter Reed, a hospital for the army. He had to classify chemical compounds according to their molecular structure. That was the project there, in medicine. And I got interested in making a contribution to this, and working it into my life's mission I found by this time. This was graph isomorphism and I was darn proud to discover a good algorithm for isomorphism of trees, a good algorithm for whether one tree is a subtree of another. I was writing up this note on tree isomorphism and putting in there my little sermon about good algorithms. And Tom Saaty took it and printed it verbatim in his book and he referred to me as originating it. There was no problem there, no problem. It is published and it contains this slogan about: Is there or is there not a polynomially or algebraically bounded algorithm for graph isomorphism? And here is one for trees and it isn't trivial. So anyway, that was a neat thing that came from my association with Lehman.

* * *

In the meantime, Al Tucker had talked to colleagues of his at the University of Maryland, I guess. And he really gave me a talk about: "Now, Jack, you get that PhD!" You know, I had been a grad student for years and I had passed all the comps. And I had done fine there, you know, on everything, but I'd just drifted away. Well, I was sore about that, the way all these PhD students were just drifting away.

And I got together with a few of the professors at their invitation and I laid it on them that it is just not good to have a graduate program like that. In fact, I was kind of hip on the idea of bestowing a title. This is a mediaeval way of the academic priests to protect their positions, right? And this promise of a PhD with all these people going away, this drifting away, as failures in it. I laid this on them and they agreed to disagree, and I said: "Shove your PhD!", you know.

And it was a very cordial thing with these individual professors that I liked. They meant a lot to me and I respected them. But just like Al Lehman, you know, Al was all hepped up about not raising his arm there and I couldn't understand the big deal about that. But this was an influence in a sense, and I felt very strongly about the oppressiveness of graduate student programs. And so, with this experience with Al at the Bureau of Standards, I just felt like it wasn't right for me to get a PhD at that point, you know.

Anyway, Ray Fulkerson and I started, after the matroid seminar, talking by long distance telephone. That was a period when I was close to him. There were various combinatorial theorems around, and his thing was — and a darned good thing — to be able to see almost every theorem under the sun as a network flow problem. And I was trying to see everything as a matroid problem. He had found this good theorem in the literature, somebody's theorem about when there is some combinatorial structure, that he had simulated as a flow problem. And he talked to me like: this seems an awful lot like matroids. And sure enough it was. Then we could also set it up as a matroid partition problem. That is in the paper Ray and I did together.

And I also got the matroid intersection theorem by this time. There was Kruskal's theorem around. There was all this flow stuff around, König's theorem, you know, all the stuff that Hoffman and so on had. That goes right over to matroids. We had got to get these two theories together, and they got to come together. And so there was matroid intersection, and I liked that actually.

But now the thing is, I had all these pieces that I was trying to fit together in the big picture here, this matroid theory. Particularly, what is it that is making this thing work? I mean these left-hand sides, there is nothing special about the matrix coefficients. But these right-hand sides, you know! They give a set function, huh? What is it about this set function? And that was kind of another eureka, you know, the submodular set functions. And gee, you know, the whole damned thing goes through with submodular set functions! Oh boy, was I on a roll!

When I started trying to get this matroid stuff together, I presented this matroid partitioning paper. And at the beginning of it I am giving this little sermon. We're going to define our basic steps in terms of recognizing independence and so on, and I give my little speech, and that has this parallel with the absolute supervisor. I stuck that in there, instead of in the matching polytope paper, about this certificate I put in there a line that, well, later on we will — I will present finding a largest independent set, you know. I talked about that, well, you've got the problem of finding an independent set of nodes in a graph and a problem of coloring the nodes with the fewest independent sets. Now here is a concept of independence where these problems work. And we will do the greedy algorithm later. And I got requests for this paper, and other people got priority on the matroids and the greedy algorithm. People that had listened to me talking about this subject and had asked for copies of my preprint on matroid partitioning, where I say: "Well, I am going to put up this other thing later", you know.

So that upset me. I mean this was a new thing, they run off in a corner and then the next time you see them, the thing is in print. And this was just like . . . , the only experience I had had like this before was when I was trying to get rich by selling toys. And a few months later you turn around and see your toy in the toy store. Well this was happening, this was happening, right and left, to these bits and pieces of this matroid theory.

At that time there were hundreds of papers of combinatorial theorems that follow from versions of network flow, versions of bipartite stuff or zero-one matrix stuff. They all are versions of max-flow equal to min-cut or such. Most of it was done in terms of these zero-one matrices.

Anyway, here was someone that had done his thesis on this kind of thing and so, every afternoon at four at coffee time, I was preaching to him about matroids. Well, a few months later he has gone and someone else had gotten his paper containing my ideas, his version of them, to referee and heard about it from me, and sent it to me. And so I came down on this guy, with a really, really bitter personal put-down on him, you know, for doing this to me. I mean, it was perfectly obvious, you know, I taught him the idea at coffee at the Bureau of Standards and he had run off in this corner to write about this.

And there was another guy, asking him about submodular functions I found out that he didn't know anything about them. And the next I hear of it is when Alan Goldman happens to go to a meeting and he comes back to tell me that this guy has presented a paper on submodular functions! And I shot off to him I guess one of the most scathing, sarcastic letters I have ever written. Because I was really, I was really strung out about . . . Bits and pieces getting away from me. I mean, this was fine. I had organized this seminar. I wanted to share. I didn't understand this running off in a corner and getting the jump. I never had any experience with that. Nothing like that at this Rand meeting, nothing like this, and my experience at Princeton, nothing like this connected with matchings. I mean, Dantzig calling me up and saying: "Hey, can my favorite student come work with you?" And, and yet.

And in this direction, on two or three guys I have explicit concrete evidence, you know. I have sent them my paper. They have heard me give these lectures about matroids and the greedy algorithm. And they are publishing these papers on it. And I am trying to get all this stuff together to a big theory, you know. And I've got to prove to this lousy academic community, you know, I've got to prove a thing or two. I've just told them to shove their PhD. And that is a big thing for me. The oppressiveness of this PhD thing, and now, you don't need a PhD to be a scholar, and I am really working to get together these submodular functions and the matroid intersection and the greedy algorithm. And I had all these pieces to put together, and go be Jesus in the temple.

You say that I don't give myself credit. No, I do. I am defensive and I am insecure, I know I am the biggest underachiever at Oberwolfach. In that sense I know that. Let me finish. As far as I am concerned, that theory that I worked out in the sixties is at least as important as what Einstein did, and I am ready to be Einstein, you know, I am ready to sit back and pontificate. I felt that way at that time, and I feel that way now. Look, this is the big moment, I mean this thing is incredible, this theory.

Oh, so, evidently, something got around somewhere because Richard Guy and Eric Milner had invited me – this was out of the blue –, but they invited me to give two weeks of a lecture a day at this international conference on combinatorial structures in Calgary in 1969. And so, this was my moment. All these years since the matroid seminar, people grabbing pieces here and grabbing pieces here and grabbing pieces there and I was bitter about this.

But now, you know, this is my day in the sun.

This text is based on an interview of Jack Edmonds by George Nemhauser in Oberwolfach, January 17, 1991 (ending the early hours of the next day), recorded by Jan Karel Lenstra. The text was compiled by Bert Gerards, Lex Schrijver and Bruce Shepherd.

Early Integer Programming

Ralph E. Gomory

During the academic year 1953-54, I was a third year graduate student in mathematics at Princeton, doing research on nonlinear differential equations. I wrote several papers, the first of which became my PhD thesis. I was fortunate in having as my thesis advisor the remarkable and inspiring Professor Solomon Lefschetz. Armed with my PhD degree, I entered the US Navy in the fall of 1954 and spent four months at Officer Candidate School in Newport, Rhode Island. After that the Navy assigned me to the Physics Branch of the Office of Naval Research in Washington, and I arrived there early in 1955.

Down the hall from the Physics Branch was the Operations Research Group. By 1956 I knew some of the people there and had learned something about what operations research was. Also by that time I had learned my duties at the Physics Branch well enough so that I did them in less than full time, and Frank Isaakson, my helpful Branch Head, permitted me to spend my spare time with the Operations Research Group.

I had always wanted to try applied mathematical work, and the time I spent with the Operations Research Group looking at various Navy weapons systems strengthened that interest. I decided tentatively to make operations research my future work, and by way of preparation took, in 1957, an evening course in operations research given by Alan Goldman. This was my first encounter with linear programming.

Later in 1957, as the end of my three year tour of duty in the Navy was approaching, Princeton invited me to return as Higgins Lecturer in Mathematics. Because of my interest in applied work I had planned to look for an industrial position, but I decided instead to accept this attractive offer and spend a year or two at Princeton before going on.