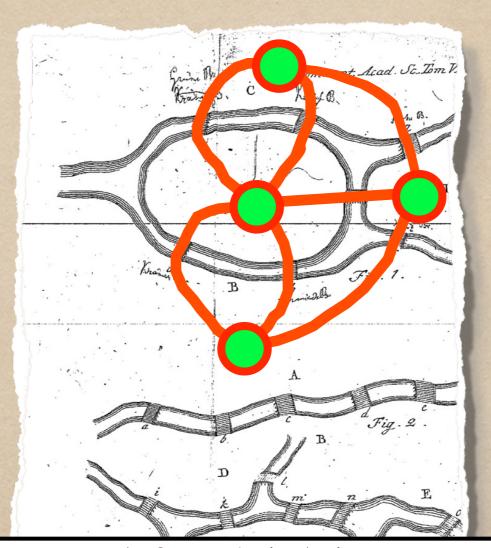


Kapitel 2.3: Eulerwege

Algorithmen und Datenstrukturen WS 2024/25

Prof. Dr. Sándor Fekete

2.1 Historie

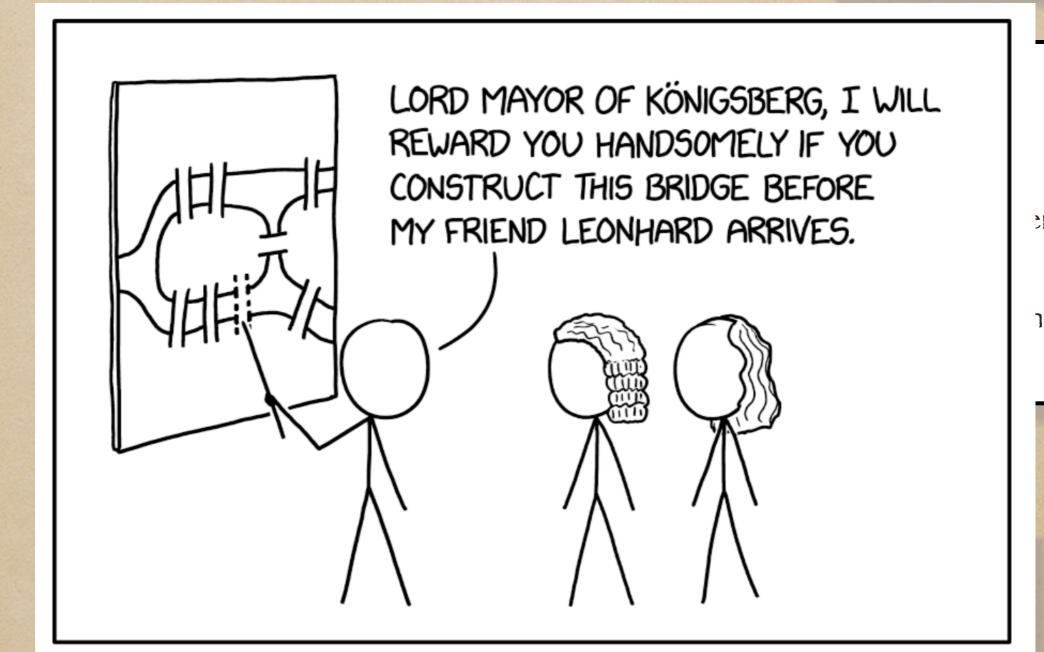


- Alle Knoten sind ungerade?!
- Man müsste an allen anfangen oder aufhören!
- Das geht nicht an einem Stück!

Euler: (1) Das gilt für jede beliebige Instanz: Mit mehr als zwei ungeraden Knoten gibt es keinen solchen Weg.

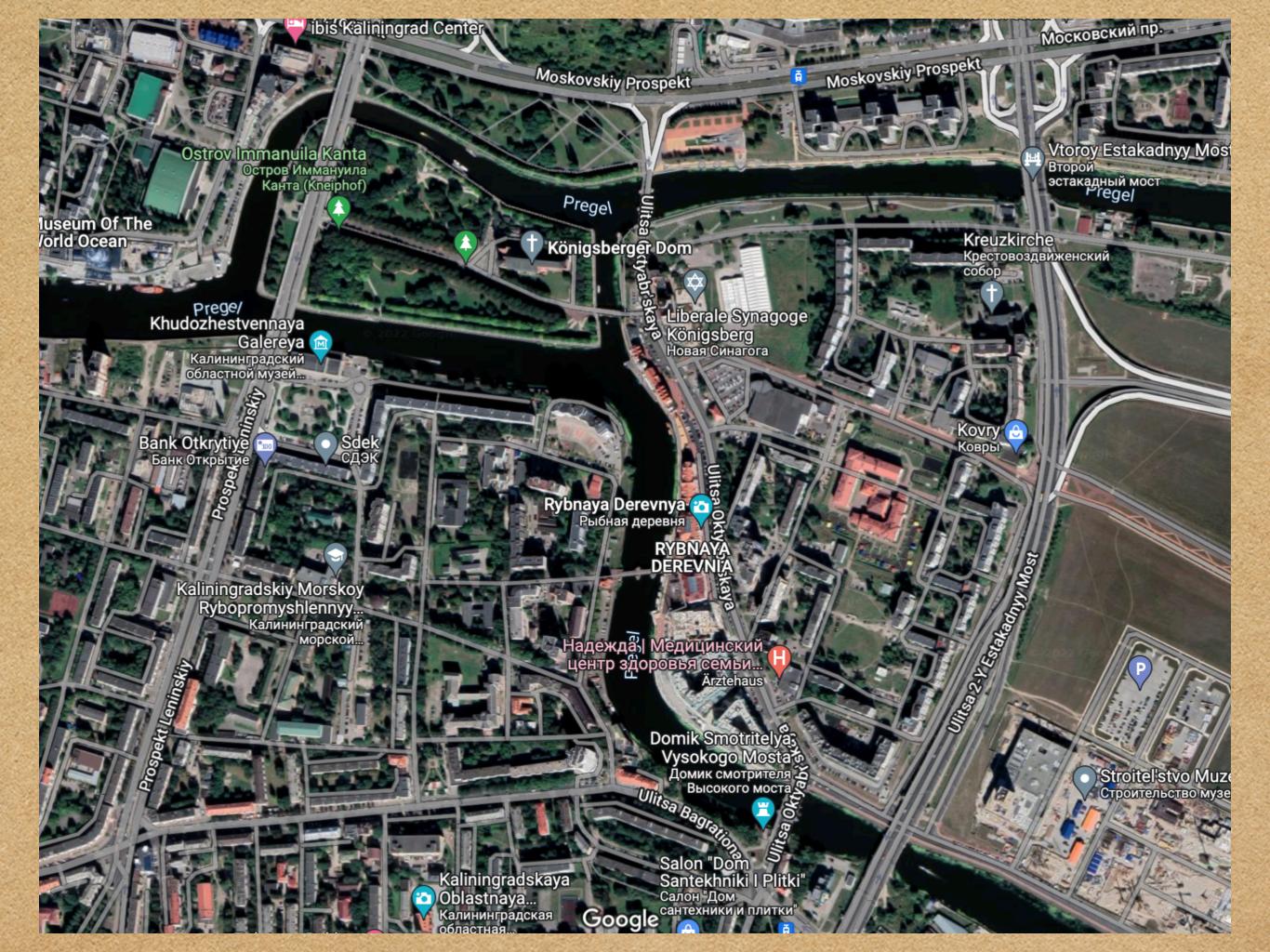
(2) Man kann auch charakterisieren, unter welchen Bedingungen es einen Weg tatsächlich gibt.

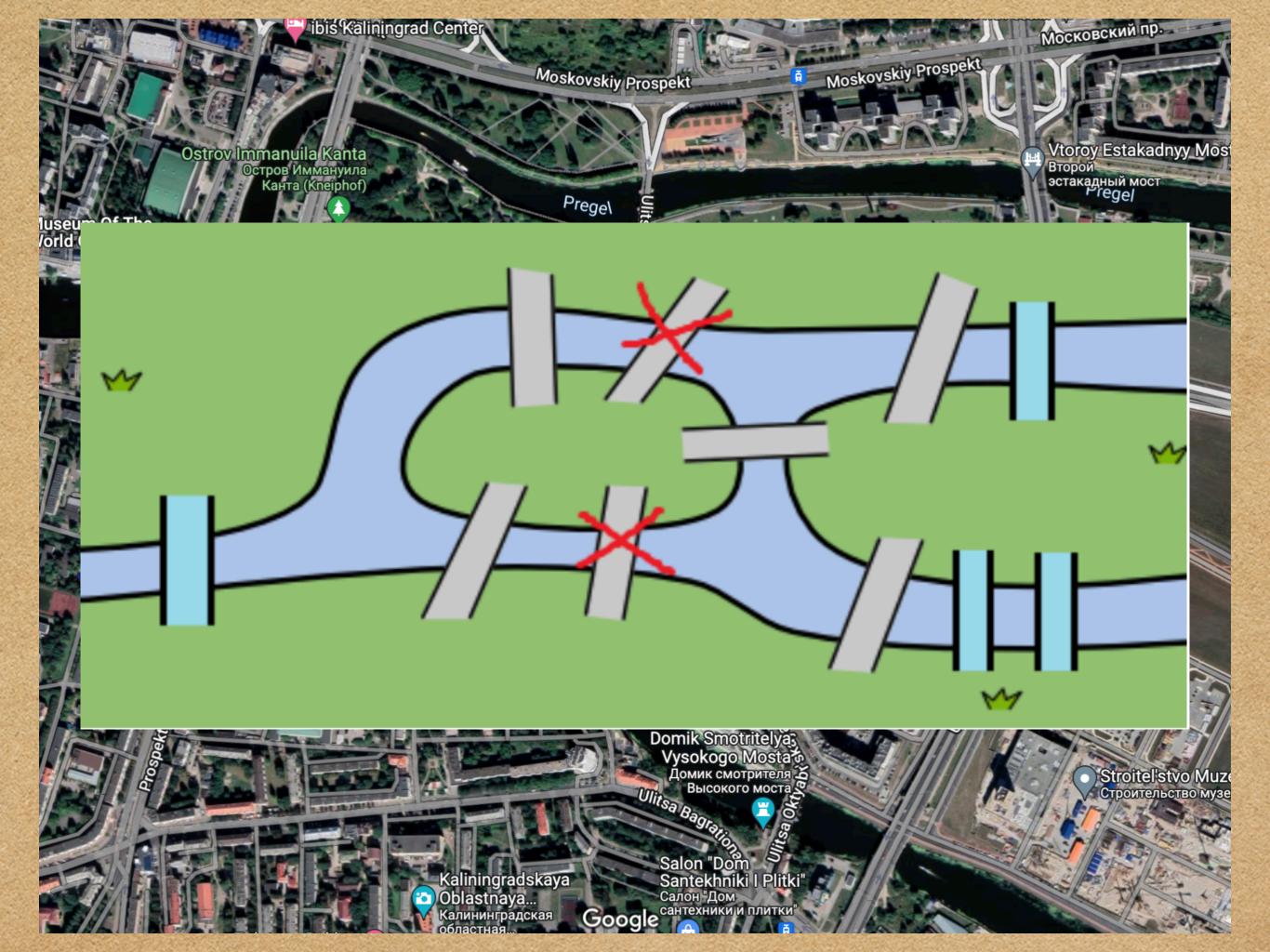
2.1 Historie



I TRIED TO USE A TIME MACHINE TO CHEAT ON MY ALGORITHMS FINAL BY PREVENTING GRAPH THEORY FROM BEING INVENTED.

7





Problem 2.3 (Eulerweg)

Gegeben: Ein Graph G=(V,E)

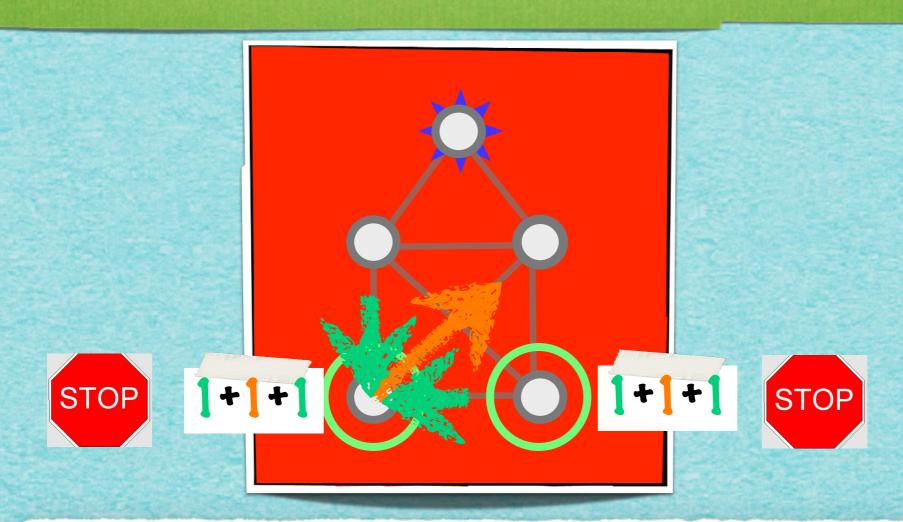
Gesucht: Ein Eulerweg W in G - oder ein Argument, dass

kein Eulerweg existiert

Satz 2.4 (Euler)

- (1) Ein Graph G=(V,E) kann nur dann einen Eulerweg haben, wenn es höchstens zwei Konten mit ungeradem Grad gibt.
- (2) Ein Graph G=(V,E) kann nur dann einen geschlossenen Eulerweg haben, wenn alle Konten geraden Grad haben.

Das Haus des Nikolaus



Beweis. Seien G=(V,E) ein Graph und $v\in V$ ein Knoten mit ungeradem Grad $\delta(v)$. Dann kann die Zahl der in einem Eulerweg W zu v hinführenden Kanten nicht gleich der von v wegführenden Kanten sein. Also muss W in v beginnen oder enden. Damit:

- (1) Es gibt in einem Eulerweg nur einen Start- und Endknoten.
- (2) Bei einem geschlossenen Eulerweg gibt es für den Start- und Endknoten w gleich viele hin- und wegführende Kanten. Also ist auch $\delta(w)$ gerade.

Fragen:

(I) Was ist mit Graphen, in denen nur ein Knoten ungeraden Grad hat?
 (II) Die Bedingungen oben sind notwendig, d.h. sie müssen auf jeden Fall erfüllt werden, wenn es eine Chance auf einen Eulerweg geben soll. Sind sie auch hinreichend, d.h. gibt es bei Erfüllung auch wirklich

einen Eulerweg?

(III) Wie findet man einen Eulerweg?

(III') Wie sieht ein Algorithmus dafür aus?

Zu (I):

Satz 2.5 (Handshake-Lemma)

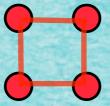
Für jeden einfachen Graphen ist die Zahl der Knoten mit ungeradem Grad eine gerade Zahl.

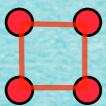
Zu (II):

Gegeben ein Graph G

- mit nur zwei ungeraden Knoten. Hat er einen Eulerweg?
- mit nur geraden Knoten. Hat er eine Eulertour?

Nein!





Der Graph muss zusamenhängend sein!

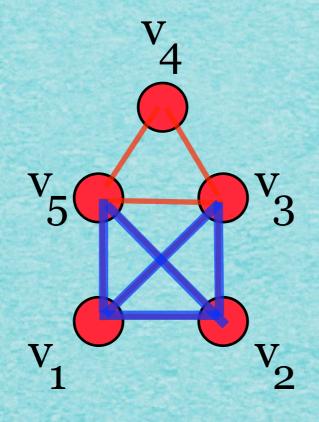
Zu (II):

Gegeben ein zusammenhängender Graph G

- mit nur zwei ungeraden Knoten. Hat er einen Eulerweg?
- mit nur geraden Knoten. Hat er eine Eulertour?

Grundtechnik:

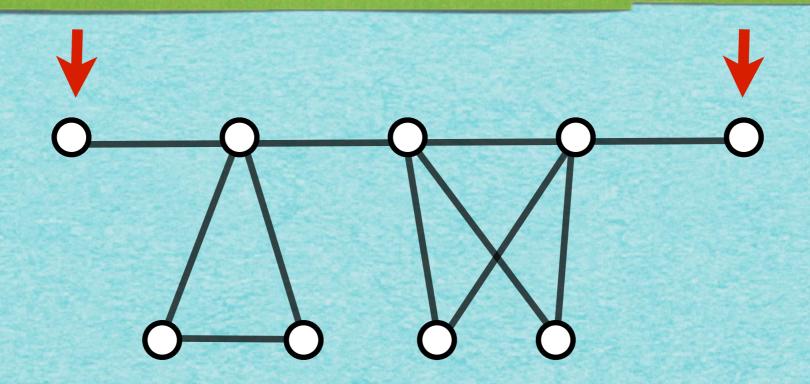
Um zu sehen, wo man etwas richtig machen muss, überlegt man sich, wo man etwas falsch machen kann!

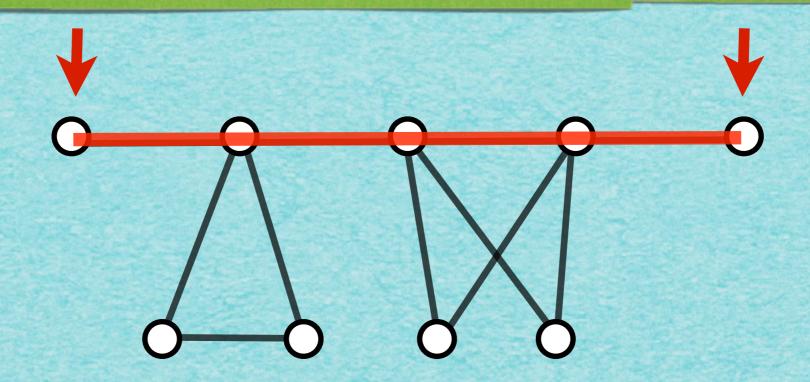


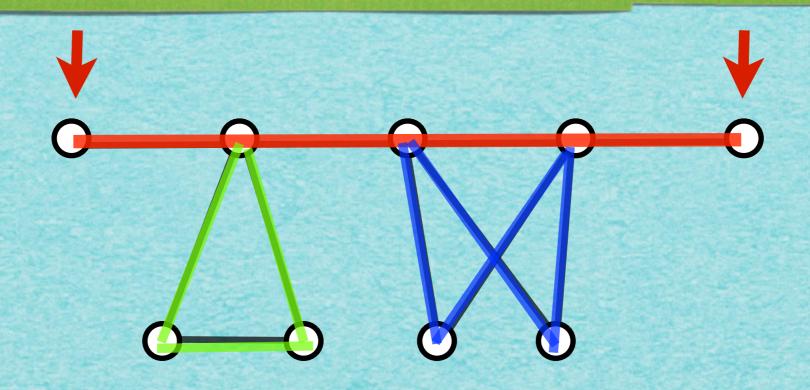
Wir hinterlassen Kanten?!

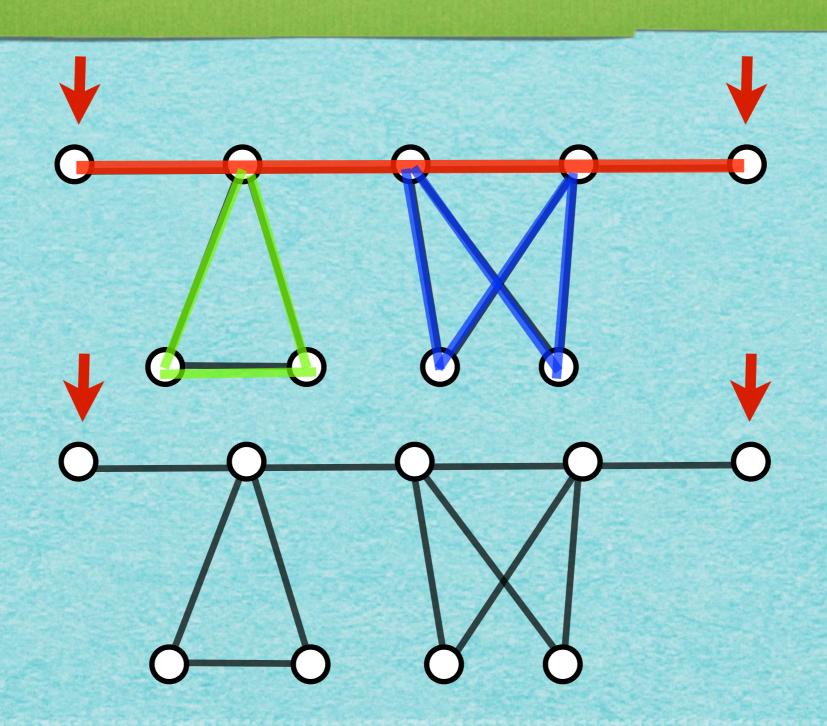
"Vorlesung"

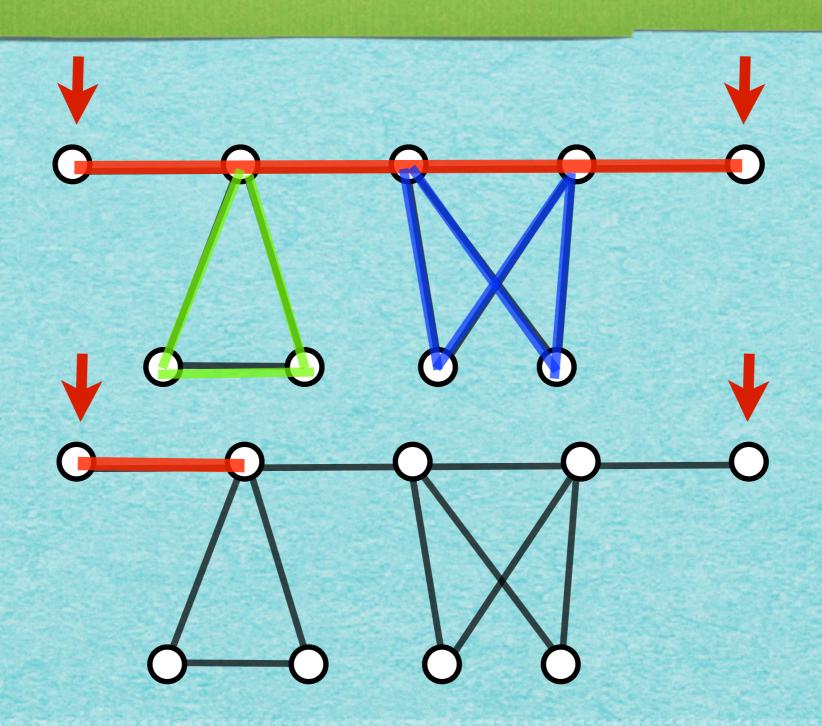


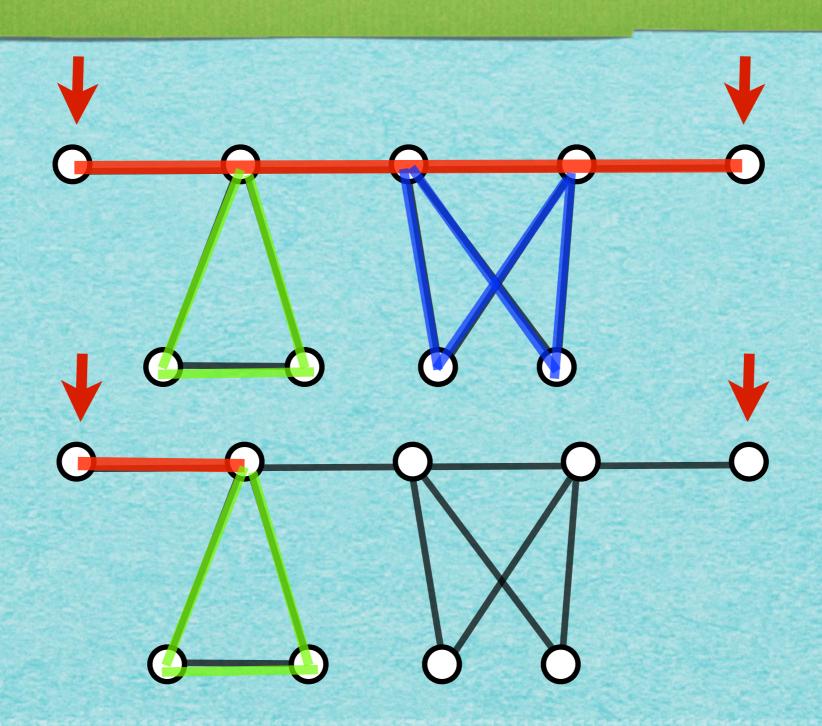


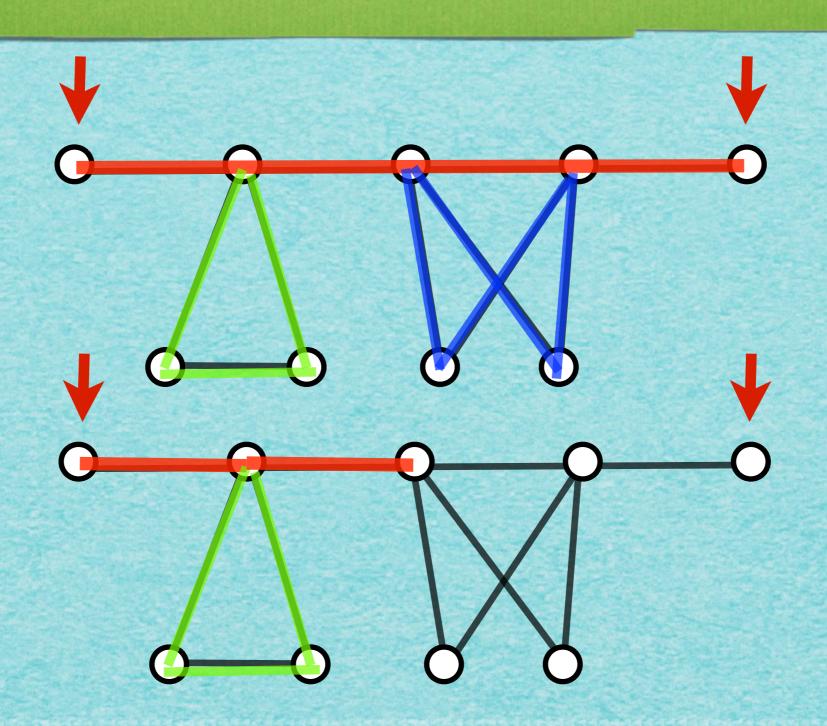


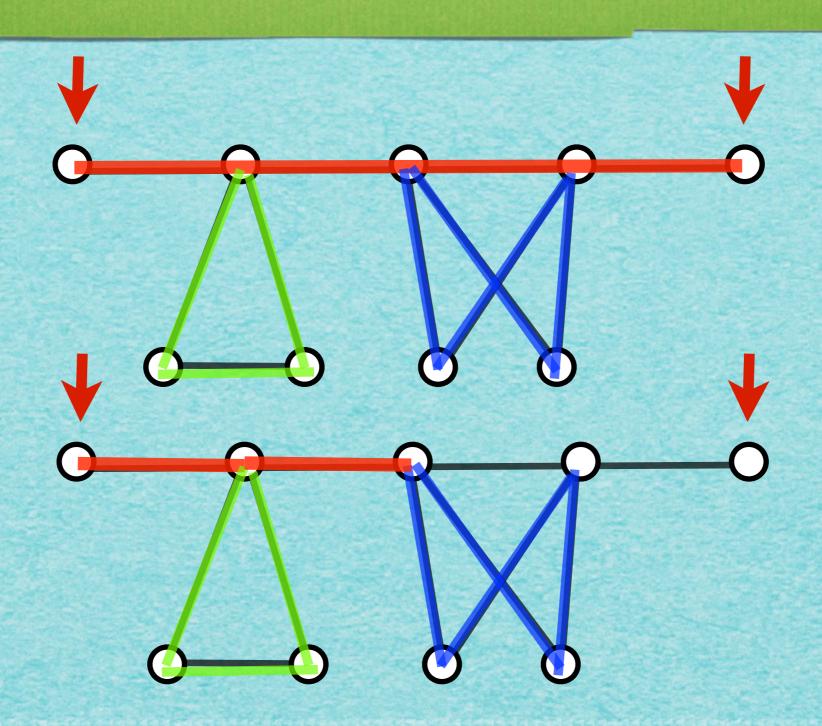


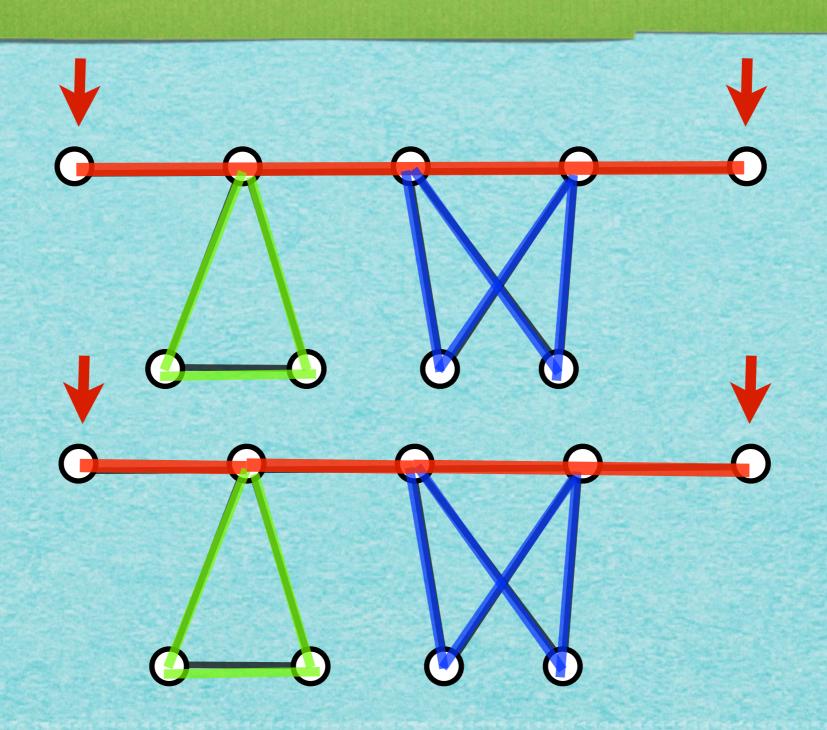












128 SOLF

SOLVTIO PROBLEMATIS

SOLVTIO PROBLEMATIS

AD

GEOMETRIAM SITVS

PERTINENTIS.

AVCTORE

Leonb. Eulero.

Ģ. г.

Tabula VIII. Raeter illam Geometriae partem, quae circa quantitates versatur, et omni tempore summo studio est exculta, alterius partis etiamnum admodum ignotae primus mentionem fecit Leibnitzius, quam Geometriam fitus vocauit. Ista pars ab ipso in solo fitu determinando, fitusque proprietatibus eruendis occupata effe statuitur; in quo negotio neque ad quantitates respiciendum, neque calculo quantitatum vrendum sit. Cuiusmodi autem problemata ad hanc fitus Geometriam pertineant, et quali methodo in iis resoluendis vti oporteat, non fatis est definitum. Quamobrem, cum nuper problematis cuiusdam mentio esset facta, quod quidem ad geometriam pertinere videbatur, at ita erat comparatum, vt neque determinationem quantitatum requireret, neque folutionem calculi quantitatum ope admitteret, id ad geometriam fitus referre haud dubitaui: praesertim quod in eius solutione solus situs in considerationem vemat, calculus vero nullius prorfus fit vius. Methodum ergo meam quam ad huius generis proble-

Euler: (1) Das gilt für jede beliebige Instanz: Mit mehr als zwei ungeraden Knoten gibt es keinen solchen Weg.

(2) Man kann auch charakterisieren, unter welchen Bedingungen es einen Weg tatsächlich gibt. Hierholzer proved that a graph has an Eulerian cycle if and only if it is connected and every vertex has an even degree (excluding the starting and terminal vertices). This result had been given, without proof, by Leonhard Euler in 1736. Hierholzer apparently explained his proof, just before his premature death in 1871, to a colleague who then arranged for its posthumous publication which appeared in 1873.^[1]

Mathematische Annalen

March 1873, Volume 6, <u>Issue 1</u>, pp 30–32

Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren.

Von CARL HIERHOLZER.
Mitgetheilt von CHR. WIENER*).

In einem beliebig verschlungenen Linienzuge mögen Zweige eines Punktes diejenigen verschiedenen Theile des Zuges heissen, auf welchen man den fraglichen Punkt verlassen kann. Ein Punkt mit mehreren Zweigen heisse ein Knotenpunkt, der so vielfach genannt werde, als

^{*)} Die folgende Untersuchung trug der leider so früh dem Dienste der Wissenschaft durch den Tod entrissene Privatdocent Dr. Hierholzer dahier (gest. 13. Sept. 1871) einem Kreise befreundeter Mathematiker vor. Um sie vor Vergessenheit zu bewahren, musste sie bei dem Mangel jeder schriftlichen Aufzeichnung aus dem Gedächtniss wieder hergestellt werden, was ich unter Beihilfe meines verehrten Collegen Lüroth durch das Folgende möglichst getren auszuführen suchte.

Algorithmus 2.7

INPUT: Graph G mit höchstens zwei ungeraden Knoten

OUTPUT: Ein Weg in G.

- 1. Starte in einem Knoten v₀ (ungerade, sonst beliebig);
- 2. Solange es eine zum gegenwärtigen Knoten v_i inzidente unbenutzte Kante $\{v_i\,,\,v_j\}$ gibt:
 - 2.1. Wähle eine dieser Kanten aus, $e_i = \{v_i, v_j\}$
 - 2.2. Laufe zum Nachbarknoten v_j
 - 2.3. Lösche die Kante aus der Liste der unbenutzten Kanten.
 - 2.4. Setze $v_{i+1} := v_j$
 - 2.5.Setze i := i+1
- 3. STOP

Algorithmus 2.8

Algorithmus von Hierholzer

INPUT: Ein zusammenhängender Graph G mit höchstens zwei ungeraden Knoten

<u>OUTPUT:</u> Ein Eulerweg bzw. eine Eulertour in G

- A. Wähle einen Startknoten v (ungerade falls vorhanden);
- B. Verwende Algorithmus 2.7, um einen Weg W von v aus zu bestimmen;
- C. Solange es noch unbenutzte Kanten gibt:
 - C.1. Wähle einen von W besuchten Knoten w mit positivem Grad im Restgraphen;
 - C.2. Verwende Algorithmus 2.7, um einen Weg W' von w aus zu bestimmen;
 - C.3. Verschmelze W und W'
- D. STOP

Satz 2.9

- (i) Das Verfahren 2.7 stoppt immer in endlicher Zeit, ist also ein Algorithmus.
- (ii) Der Algorithmus liefert einen Weg.
- (iii) Ist v₀ ungerade, stoppt der Algorithmus im zweiten ungeraden Knoten.
- (iv) Ist G eulersch (d.h. haben alle Knoten gerade Grad), stoppt der Algorithmus in v_0 , liefert also einen geschlossenen Weg.

Beweis:

(i)

Satz 2.9

- (i) Das Verfahren 2.7 stoppt immer in endlicher Zeit, ist also ein Algorithmus.
- (ii) Der Algorithmus liefert einen Weg.
- (iii) Ist v₀ ungerade, stoppt der Algorithmus im zweiten ungeraden Knoten.
- (iv) Ist G eulersch (d.h. haben alle Knoten gerade Grad), stoppt der Algorithmus in v_0 , liefert also einen geschlossenen Weg.

Beweis:

(i)

Algorithmus 2.7

INPUT: Graph G mit höchstens zwei ungeraden Knoten

OUTPUT: Ein Weg in G.

- 1. Starte in einem Knoten v_0 (ungerade, sonst beliebig);
- 2. Solange es eine zum gegenwärtigen Knoten v_i inzidente unbenutzte Kante $\{v_i\,,\,v_j\}$ gibt:
 - 2.1. Wähle eine dieser Kanten aus, $e_i = \{v_i, v_j\}$
 - 2.2. Laufe zum Nachbarknoten vi
 - 2.3. Lösche die Kante aus der Liste der unbenutzten Kanten.
 - 2.4. Setze $v_{i+1} := v_j$
 - 2.5.Setze i := i+1
- 3. STOP

Satz 2.9

- (i) Das Verfahren 2.7 stoppt immer in endlicher Zeit, ist also ein Algorithmus.
- (ii) Der Algorithmus liefert einen Weg.
- (iii) Ist v₀ ungerade, stoppt der Algorithmus im zweiten ungeraden Knoten.
- (iv) Ist G eulersch (d.h. haben alle Knoten gerade Grad), stoppt der Algorithmus in \mathbf{v}_0 , liefert also einen geschlossenen Weg.

Beweis:

(i) Bei jedem Durchlauf der Schleifen 2.1-2.5 wird in 2.3 eine Kante entfernt. Das kann nur endlich oft passieren. Also muss das Verfahren irgendwann stoppen.

Fortsetzung folgt...

s.fekete@tu-bs.de