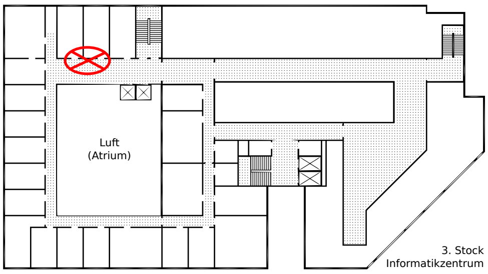
Algorithms Group **Summer '23**
Departement of Computer Science - IBR
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Dr. Ahmad Moradi

Approximation Algorithms: Homework 3 (7. June)

Solutions are due on June, 21th until 11:00 AM in the cupboard for handing in practice sheets. Please put your name on all pages.



Exercise 1 (Densest Subgraph):
Given a graph $G = (V, E)$, for any subset $S \subseteq V$ define

$$den(S) = \frac{|E(S)|}{|S|}$$

so as to measure *density* of S where $E(S)$ is the set of all edges in E with both endpoints in S. In the *densest subgraph problem* we need to find a subset of nodes, say S^*, that has the maximum density. In the following, we consider a greedy idea giving a $\frac{1}{2}$-approximation factor algorithm for the problem

a) For each edge $ij \in E$ arbitarily assign the edge either to i or to j, and let d_i be the number of nodes assigned to i this way. Show that

$$\max_{S \subseteq V} den(S) \leq \max_{i \in V} d_i$$

b) Consider the following edge assignement idea where each edge assigns itself to the first incident vertex deleted by the algorithm below.

Algorithm 1: Greedy Densest Subgraph

Data: Graph $G = (V, E)$
$S, S' \leftarrow V$, ;
while $S \neq \emptyset$ **do**
  Find a minimum degree vertex in $G(S)$[1] and delete it from S ;
  if $den(S) > den(S')$ **then**
    $S' \leftarrow S$;
return S';

[1] $G(S)$ is the subgraph induced by S, i.e. $G(S) = (S, E(S))$

Show that

$$\max_{i \in V} d_i \leq 2\ den(S')$$

where S' is the output of the algorithm above and deduce that the algorithm outputs a $\frac{1}{2}$-approximation to the problem.

(10+10 P.)

Exercise 2 (Integrality Gap):
As we already discussed in previous lecture, in order for Linear Programming (LP) rounding algorithms to work properly, we need that our LP optimal value, OPT_{LP}, to be not too far away from actual optimal value, OPT. It is then interesting to ask in other words: what is the best approximation ratio we can expect if we use the LP optimal value as the lower bound? The notion of *integrality gap*, provides a way to study this question.

Definition The integrality gap of a linear program for a minimization problem Π is

$$\sup_{\text{instance } I \text{ of } \Pi} \frac{OPT(I)}{OPT_{LP}(I)}$$

Now, try to answer the following questions:

a) Given a minimization problem Π, discuss what information is provided when integrality gap of an LP is known.

b) Consider minimum weighted vertex cover problem and the ILP formulation discussed in the last lecture. Prove that integrality gap of the LP relaxation provided is at least $2 - \frac{1}{n}$, where n is the number of nodes in the input instance.

   Hint. Consider a complete graph, $G = K_n$ as input instance.

c) Given a grpah $G = (V, E)$, an independent set of the graph is a subset of nodes amonge which no edge exists. Here, we are looking for the maximum size independent set in G. Observe that $I \subseteq V$ is an independet set if and only if $V \setminus I$ is a vertex cover. Modify the ILP formulation for vertex cover to obtain a formulation for independet set problem. Prove that the integrality gap of this problem is at least $\frac{n}{2}$ where n is the number of nodes in the input instance.

d) As mentioned above, konwing an optimal solution to vertex cover gives an optimal independent set and vice versa. Knowing that, what could be infered from this huge difference between their integrality gaps. Discuss.

(5+10+5+10+5 P.)