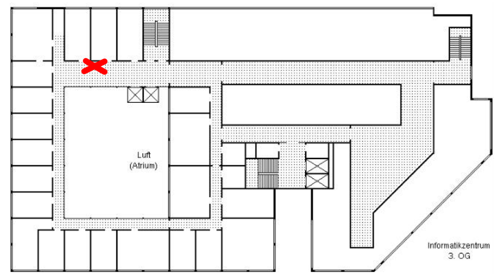
Abteilung Algorithmik
Institut für Betriebssysteme
und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor Fekete
Dr. Christiane Schmidt

SS 12

Approximation Algorithms
Homework Set 4, 13. 06. 2012

Solutions are due Wednesday, June 27th, 2012, until 13:00 in the cupboard for handing in practice sheets. **Please put your name on all pages!**



Exercise 1 (MAX CUT):
We consider the problem MAX CUT:
Input: an undirected graph $G = (V, E)$ with vertex set V and edge set E.
Output: a partition $(S, V \setminus S)$ of the vertex set, such that the size $w(S)$ of the cut, that is, the number of edges between S and $V \setminus S$, is maximized.

(a) Consider the example graph G from Figure 1. Give a MAX CUT S for G. What is its size?

The problem MAX CUT is NP-hard, hence, we consider the following approximation algorithm:

Algorithm

```
S = ∅
while ∃v ∈ V : w(SΔ{v}) > w(S) do
        S = SΔ{v}
return S
```

Here, Δ gives the symmetric difference of two sets, so:

$$S\Delta\{v\} = \begin{cases} S \cup \{v\} & : \quad v \notin S \\ S \setminus \{v\} & : \quad \text{otherwise} \end{cases}$$

So our algorithms starts with a vertex set S and as long as there exists a vertex that if added or deleted from S increases the current cut, S is adapted accordingly (with a local improvement).

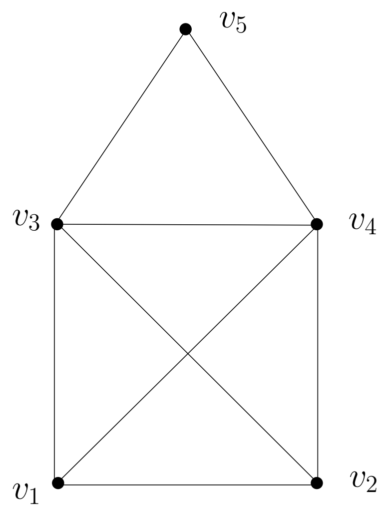


Abbildung 1: Graph G.

(b) Apply the algorithm to the graph H from Figure 2. In case of ties use the following rule: prefer adding vertices over deleting vertices; in case there still is a tie, use the vertex with the smallest index.

(c) Show: for every given input the algorithm outputs a cut of size $w \geq \frac{1}{2}OPT$, where OPT denotes the size of an optimal cut.

(d) Show that the algorithm has polynomial running time.

(e) Was the analysis from (c) best possible? That is, is there a graph $G = (V, E)$, such that the algorithm finds a feasible solution $S \subseteq V$ with $w(S) = \frac{1}{2} \cdot OPT(G)$? (Give a graph with an arbitrary number of nodes.)

(5+10+10+7+10 Punkte)

Exercise 2 (Bin Packing II):
Consider another algorithm for MIN BIN PACKING: the next fit algorithm. At each step there is exactly one open bin B_j. The next item is packed into B_j if it fits, otherwise, a new bin B_{j+1} gets opened, B_j gets closed and will never be opened again.

(a) Show that the next fit algorithm has an approximation factor of 2.

(b) Show that the bound from (a) cannot be improved.

(10+8 Punkte)

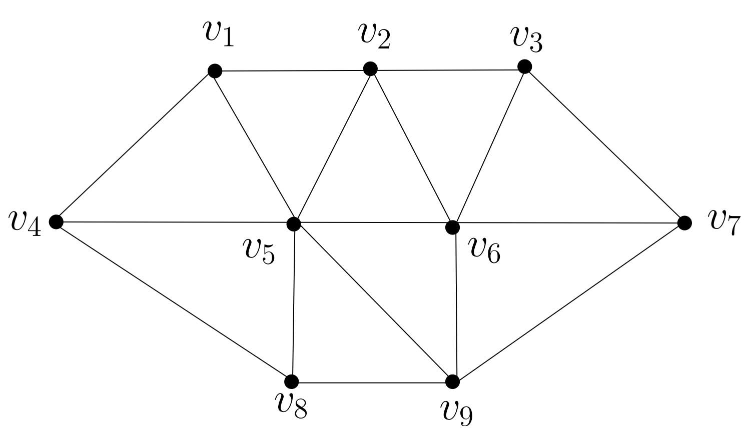


Abbildung 2: Graph H.