Diskrete Mathematik II — WS 2012 / 2013
Tibor Szabó

Exercise Sheet 6

Due date: Nov 27, 12:30 PM, beginning of exercises
NO LATE SUBMISSIONS!

You should try to solve and write up all the exercises. You are welcome to submit **at most** two neatly written exercises each week. You are encouraged to submit in pairs, but don't forget to mark the name of the scriber.

Exercise 1 (The menu).
A new restaurant has opened and the manager is trying to decide the daily menu offered by the restaurant. For this purpose he has created a list $\mathcal{L} = \{D_1, \ldots, D_n\}$ of possible dishes. He knows that the preparation of dish D_i requires a certain set of ingredients $\mathcal{A}_i \subseteq \mathcal{A}$. Here $\mathcal{A} = \{I_1, \ldots, I_m\}$ represents the (finite) list of possible ingredients. It is possible that one ingredient is required by more than one dish. The manager has computed (based on statistical data) the income y_i obtained by offering dish D_i on the daily menu, as well as the cost x_j of buying ingredient I_j, $1 \leq i \leq n, 1 \leq j \leq m$. It is assumed for simplicity that once an ingredient is bought, it can be used for any number of dishes.

The manager would like to determine a daily menu that maximizes the net revenue, i.e. the total income from the dishes offered minus the cost of the ingredients necessary for these dishes. Please describe an efficient algorithm that finds a solution to this problem.

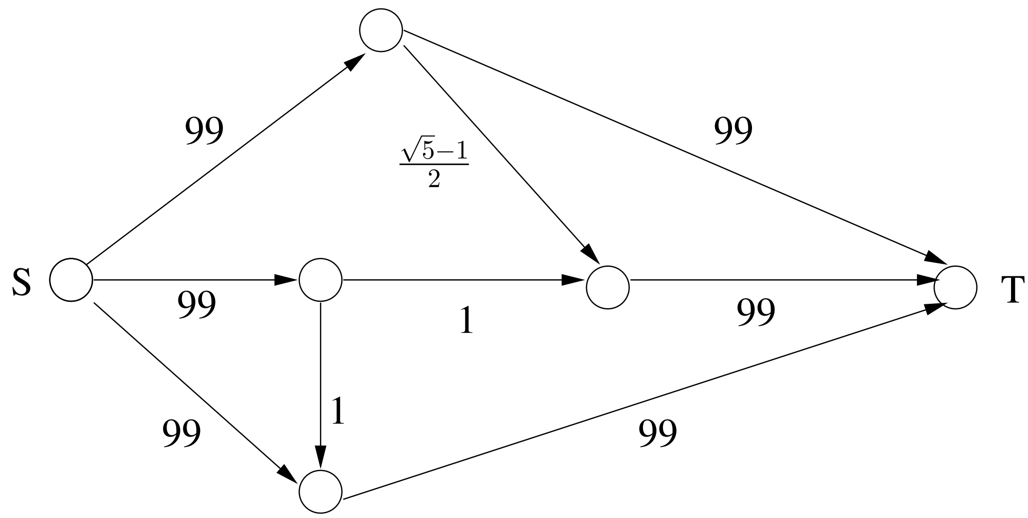
Exercise 2 (Path decomposition).
Suppose G is a directed acyclic graph on n vertices. A **path decomposition** of G is a collection of vertex-disjoint directed paths in G such that every vertex of G is contained in exactly one path of the collection. Paths are allowed to have length 0, i.e. they may contain only one vertex. The size of the path decomposition is the number of paths in the collection. Describe an efficient algorithm that finds a path decomposition of G of minimum size.

(*Hint*: Suppose p is the minimum size of a path decomposition. Construct a capacity network in such a way that the maximum flow through the network equals $n - p$).

Exercise 3 (Kneser graphs).
For $n \geq k \geq 1$, we define the *Kneser graph* $K(n, k)$ as the graph on vertex set $\binom{[n]}{k}$, and where two vertices are adjacent iff the corresponding sets are disjoint. It is a long-standing open problem to show that for $n \geq 2k + 1$, $K(n, k)$ has a hamiltonian cycle. Prove that $K(n, k)$ has a hamiltonian cycle if $n \geq \max\{(k + 1)k, 3\}$ and $k|n$.

Exercise 4.

Consider the capacity network in the figure below. The source and the sink are marked with S and T, and the capacity of every edge is indicated.



(a) Find (with arguments) the value of the maximum flow in the network.

(b) Describe a choice of augmenting paths in the Ford-Fulkerson algorithm for which the algorithm never finishes and the flow value converges to $2 + \sqrt{5}$.