Exercise Sheet 5

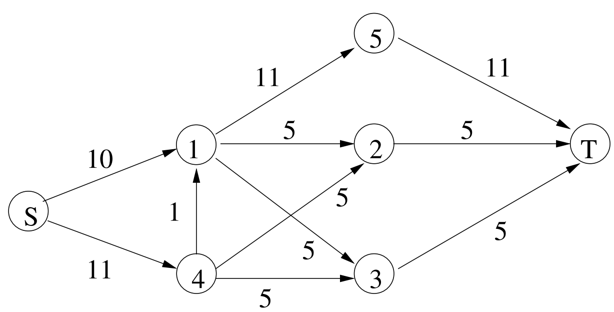
Due date: Nov 20, 12:30 PM, beginning of exercises
NO LATE SUBMISSIONS!

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.
Let G be an undirected graph. Recall that for any two distinct vertices $x, y \in V(G)$ we denote by $\lambda(x, y)$ the maximum number of pairwise internally vertex-disjoint paths between x and y.

Let $A := \{a_1, \dots, a_p\}$ and $B := \{b_1, \dots, b_q\}$ be disjoint sets of vertices of G such that $\lambda(a_i, b_j) \geq k, 1 \leq i \leq p, 1 \leq j \leq q$. Let $\nu_1, \dots, \nu_p$, $\mu_1, \dots, \mu_q$ be non-negative integers with $\sum_{i=1}^{p} \nu_i = \sum_{i=1}^{q} \mu_i = k$. Prove that there exists k independent $A - B$ paths in G such that ν_i of those start at a_i and μ_j of those end at b_j, $1 \leq i \leq p, 1 \leq j \leq q$.

Exercise 2.
Consider the directed graph in the figure.



The values on the arcs represent the capacities. The task is to compute a maximum flow from S to T using the Ford-Fulkerson algorithm. Recall that at each iteration of the algorithm, an augmenting path is found in the flow network and the flow is modified along the arcs of the path. You must write down for each iteration the augmenting path found and the value by which the flow increases.

In order to ensure uniqueness, if at some iteration there are several available augmenting paths, you must choose the one having minimum label of a vertex from the path (only the labels $\{1, 2, \ldots, 5\}$ are considered) as small as possible. If there are still several choices available, you must choose the path having second smallest label as small as possible, and so on. For example, between $[S, 4, 2, T]$ and $[S, 4, 3, T]$ you must choose the first path, as it has minimum label 2, while the second one has minimum label 3.

Exercise 3.

Let $k \geq 2$. Prove that if G is a k-connected graph on $n \geq k + 1$ vertices then for any choice of k vertices from G, there is always a simple cycle containing all of them.

Exercise 4.

Consider the following problem.

We are given an $n \times n$ black and white checkerboard with several pawns already placed on it. We would like to place as many rooks as possible on **free, white squares** of the board such that no two rooks attack each other. (A square is *free* if it is not occupied by a pawn. Two rooks *attack* each other if their positions are connected by a vertical or horizontal segment of free squares. That is, a rook does not attack another one if there is a pawn between them.)

Please describe an efficient algorithm that, given the $n \times n$ checkerboard with the pawn placement, finds a way to place a maximum possible number of rooks.