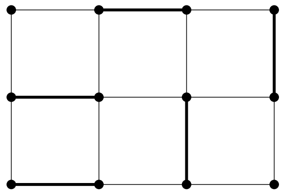
Mathematical Methods for Computer Science I

Fall 2020

Series 7 – Hand in before Monday, 9.11.2020 - 12.00

1. a) How many different maximal matchings does the complete bipartite graph $K_{m,n}$ have? (Without loss of generality, you can assume $m \leq n$.)
   b) How many different perfect matchings does the complete graph K_{2n} have?

2. a) Find an augmenting path for the matching shown below. Use this path to modify the matching to one with more edges.



   b) A graph G is called *Hamiltonian* if it has a *Hamiltonian cycle*, that is, a cycle that contains all vertices of G. Show that every Hamiltonian graph with an even number of vertices has a perfect matching.

The final three exercises deal with matchings of weighted graphs. Let $G = (V, E)$ be a graph, and let $w : E \to \mathbb{R}_{\geq 0}$ be a weight function on the edges. The goal is to find matchings $M \subset E$ so that the sum of the edge weights is large. We study the following greedy algorithm for finding an inclusion-maximal matching (similar to Kruskal's algorithm):

i) order the edges with decreasing weights: $(e_1, \ldots e_m)$ with $w(e_1) \geq \cdots \geq w(e_m)$,
ii) let M_0 be the empty set,
iii) having defined M_i, let $M_{i+1} = M_i \cup \{e_{i+1}\}$ if e_{i+1} has no common vertex with the edges in M_i, and $M_{i+1} = M_i$ otherwise.

The output of this greedy algorithm is $M_G := M_m$.

3. Choose any weight function $w : E \to \mathbb{R}_{\geq 0}$ on the edges of the Petersen graph, and carry out the above greedy algorithm.

4. Let $v_w(G)$ be the maximal possible value of the sum of edge weights of a matching in G. Show that the sum of edge weights of the matching M_G obtained by the greedy algorithm described above is at least $\frac{1}{2}v_w(G)$.

   (*Hint*: Let M be any matching of G. To each edge $e \in M$, assign the first edge of M_G having a common endpoint. In this way, each edge $e \in M_G$ is assigned to at most two edges $e_1, e_2 \in M$, with $w(e_1), w(e_2) \leq w(e)$.)

5. Show that the bound in Exercise 4 is optimal; that is, for any constant $\alpha > \frac{1}{2}$, there exists an input (G, w) for which the greedy algorithm described above finds a matching with sum of edge weights smaller than $\alpha v_w(G)$.

Remark: There exist algorithms that produce matchings of maximal combined edge weight $v_w(G)$. These are more elaborate than the greedy algorithm we have considered.