Homework set 2

Due: Friday, June 6 by 1pm– submit on Thur in class or on Friday by leaving your homework in the homework dropbox in the CS office. Solve any five of the six given problems.

1. Below is a graph G given by its adjacency. Apply the 1.5-approximation algorithm given in class to the TSP problem for the graph G below. State the upper bound on the TSP problem the approximation algorithm gives in this case, as well as the TSP cycle you obtain with this algorithm.

0	8	10	8	6	4
	0	12	9	7	6
		0	10	11	8
			0	7	5
				0	3
					0

2. Let G be a complete undirected graph in which all edge lengths are either 1 or 2 (clearly, G satisfies the triangle inequality). Give a $\frac{4}{3}$ factor algorithm for TSP in this special class of graphs. **Hint:** Start by finding a minimum 2-matching of G. A 2-matching is a subset S of edges so that every vertex has exactly 2 edges of S incident at it.

3. Let $G = (V, E)$ be a graph with edge costs satisfying the triangle inequality , and $V' \subseteq V$ be a set of even cardinality. Prove or disprove: The cost of a minimum perfect matching on V' is bounded above by the cost of a minimum cost perfect matching of V.

4. Consider a variant of the maximum-coverage problem where every element u of the universe U has to be covered at least t_u times in order to be considered covered. Let $F(S)$ be the number of covered elements (where coverage is defined as above) when the sets in S are chosen. Show that $F(S)$ is not submodular.

5. Consider a graph $G = (V, E)$. Now for every pair of nodes u, v let $S_{u,v}$ be the shortest path from u to v (if there are multiple shortest paths randomly pick one of them). Design an algorithm for the following problem: Pick k nodes from V such that the number of pairs (u, v) whose shortest paths $S_{u,v}$ go through the picked nodes are maximized. Prove the approximation factor of this algorithm. Write the pseudocode of your algorithm and compute its computational complexity.

6. Given an undirected graph $G = (V, E)$, the cardinality maximum cut problem (CMCP) asks for a partition of V into sets A and B so that the number of edges running between these sets is maximized. Consider the `CMCP-greedy` algorithm below for this problem. Show that this is a $\frac{1}{2}$-approximation algorithm.

Algorithm 1 `CMCP-greedy` algorithm

Input: graph $G = (V, E)$

$A = \{v_1\}$
$B = \{v_2\}$
for $v \in V \setminus \{v_1, v_2\}$ **do**
    if $d(v, A) \geq d(v, b)$ **then**
        $B = B \cup \{v\}$
    else $A = A \cup \{v\}$
    end if
end for
return A, B