Introduction to approximation and randomized algorithms

1st home assignment

Deadline: 27th November 2019 11:59PM
You can hand your solutions in during the exercise sessions or send them to me by email to `mberg@kam.mff.cuni.cz`.

Exercise 1 (4 points)**.** Consider 10 tokens with values from 1 to 10. If we split them up into two groups of odd and even values, then the odd valued group has an average 5 and the even valued group has an average 6.

1. Can we regroup these tokens in such a way that the average of both groups increases?
2. Can we regroup these tokens in such a way that the average of both groups grater than 5.5?

Exercise 2 (8 points)**.** Let us look at the problem of Steiner tree. We are given a connected undirected graph $G = (V,E)$, cost of edges c_e and set $R \subseteq V$ of terminals. Note, that cost of edges DOES NOT have to satisfy triangle inequality. We want to find a set of edges $F \subseteq E$ of minimal cost, such that all terminals R are in the same connected component of graph (V,F). Find a 2-approximation algorithm for this problem.
Hint: You can try to assume initially that the graph satisfies triangle inequality and then try to extend it also to the case when it does not.

Exercise 3 (4 points)**.** Prove that the Chistofides algorithm for metric TSP is not better than 1.5-approximation. That means construct a class of graphs $\{G_i | i \in \mathbb{N}\}$ such that $|V(G_{i+1})| > |V(G_i)|$ and for every $\varepsilon > 0$ there exists G_i the approximation ratio on this graph is worse than $1.5 - \varepsilon$.

Exercise 4 (4 points)**.** Let us look at the problem of asymmetric TSP. We are given a directed graph $G = (V,E)$ with positive edge lengths ℓ_e. These edge lengths satisfy triangle inequality, but not necessarily symmetry. That means that $d(u,v)$ is not necessarily the same as $d(v,u)$. Show that the algorithm that goes around the minimum spanning tree is not c-approximation for any constant c. We are looking for the minimum spanning tree in a graph where we forget edge directions.
Hint: Try to construct a graph which have all edges of length 1 except for k special edges which are long. Then show that the optimal solution uses only one of those long edges, but the algorithm will use all of them.

Exercise 5 (bonus, 4 points)**.** Let us consider the minimum vertex cover problem. On input we get an undirected graph $G = (V,E)$. We want to find a set $U \subseteq V$ of minimal size such that each edge from E is incident with at least one vertex from U. One of the ways of constructing approximation algorithms is rounding of linear relaxation. First, formulate this problem as an integer program (we did this during the first exercise session) and then convert it into a linear relaxation. That means replace all constraints of the form $x \in \{0,1\}$ by constraint $x \in [0,1]$ (even constraint $x \geq 0$ would be sufficient). So we get a linear program, which we can solve in a polynomial time. Now suppose we have an optimal solution of this relaxation (linear program). Find a way how to use this optimal solution of relaxation to find a vertex cover of a size not bigger than twice the size of minimum vertex cover. In other words, find a 2-approximation algorithm.