

Reminder: you may work in groups of up to three people, but must write up solutions entirely on your own. Collaboration is limited to discussing the problems – you may not look at, compare, reuse, etc. any text from anyone else in the class. Please include your list of collaborators on the first page of your submission. Many of these problems have solutions which can be found on the internet – please don’t look. You can of course use the internet (including the links provided on the course webpage) as a learning tool, but don’t go looking for solutions.

Please include proofs with all of your answers, unless stated otherwise.

1 Min-cost bounded path (50 points)

Suppose we are given a directed acyclic graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, a cost function $c : E \rightarrow \mathbb{N}$, a length function $\ell : E \rightarrow \mathbb{N}$, two nodes $s, t \in V$, and a length bound L . Our goal is to compute the minimum cost $s \rightarrow t$ path whose length is at most L . In other words, find the path P from s to t with $\sum_{e \in P} \ell(e) \leq L$ which minimizes $\sum_{e \in P} c(e)$.

- (a) (25 points) Give a pseudopolynomial-time exact algorithm for this problem (i.e., an algorithm which solves the problem but runs in time polynomial in n, m and $\max_{e \in E} c(e), \max_{e \in E} \ell(e)$).
- (b) (25 points) Using your algorithm from the previous part, give an FPTAS (a $(1+\varepsilon)$ -approximation which runs in time polynomial in the size of the instance and $1/\varepsilon$, i.e., polynomial in $n, m, \log(\max_{e \in E} c(e)), \log(\max_{e \in E} \ell(e))$, and $1/\varepsilon$). Hint: Think about “guessing” the maximum cost of any edge that appears in the optimal path.

2 Min-degree Steiner Tree (50 points)

The Min-degree Steiner tree problem is the same as the Steiner tree problem, except instead of minimizing total cost our goal is to minimize the maximum degree. More formally, we are given an undirected graph $G = (V, E)$ and a subset $D \subseteq V$ of terminals. The goal is to find a tree T in G which spans all of D (but not necessarily all of V) and which minimizes the maximum degree.

Show how to modify the local search algorithm from class for min-degree spanning tree to get a local search algorithm for min-degree Steiner tree that runs in polynomial time and returns a tree with maximum degree at most $2\Delta^* + \log n$ (where Δ^* is the maximum degree of the optimal tree). If you’re off by $+O(1)$ that’s OK, i.e., a bound of the form $2\Delta^* + \log n + c$ is OK for constant c .

Hints:

- (1) You will need to slightly redefine a u -improvement, since adding a single edge might not create a fundamental cycle anymore. What kind of structure “acts” in a way equivalent to a non-tree edge in spanning trees?
- (2) You might want to use the following structural graph theoretic result: in any tree with n nodes in which all non-leaves have degree at least d , the number of leaves is at least $\frac{d-2}{d-1}n$. You may use this without proof, although it’s a good idea to convince yourself that it’s true.

- (3) You will also have to change the potential function when analyzing the running time. How can you change it so that the same basic idea from class works?