Problems from Exam.pdf

Exercise 0.1 Let SHORTESTPATH be the problem of, given a weighted graph G and two vertices a, b, determine the length of the shortest path from a to b. Prove that any single-pass streaming algorithm which approximates SHORTESTPATH better than $\frac{5}{3}$OPT requires $\Omega(n^2)$ space.

Exercise 0.2 Let LC be *Label Cover* and DSF be *Directed Steiner Forest*.

1. Define *Min-Rep* and *Max-Rep* versions of LC.
2. Define *Min-Rep* and *Max-Rep* versions of DSF.
3. Give an approximation preserving reduction from either Min-Rep-LC or Max-Rep-LC to DSF. (Note that this shows DST is LC-hard to approximate.

Exercise 0.3 Let $f \colon \mathsf{N} \to \mathsf{R}$ be a function with $f(n) \geq n$. Prove that NSPACE$(f(n)) \subseteq$ DSPACE$(f(n)^2)$.

Exercise 0.4 In the *online set cover problem*, we are given a set U of n elements and a family of m subsets of U in advance. However we do not know which of the elements of U we need to cover in advance. Instead, an online sequence of elements $\sigma_1, \sigma_2, \ldots, \sigma_k$ arrives one by one. When an element σ_i arrives that is not already covered by the sets picked so far, we have to pick a new set $S_i \in F$ that contains σ_i.

Let OPT(σ) denote the minimal number of sets of F that can cover the elements in σ. For an online deterministic algorithm ALG, let ALG(σ) denote the number of sets chosen by ALG on the input sequence σ. The *competitive ratio* of algorithm ALG is defined as the worse case ratio of $\frac{\mathrm{ALG}(\sigma)}{\mathrm{OPT}(\sigma)}$ over all input sequences.

Prove that no online deterministic algorithm can achieve a competitive ratio $o((\log(n+m))$ (even with no bound on the running time).

Hint: For every $m \geq 1$, one can construct and instance with $|U| = 2^m$, $|F| = m$, such that any online deterministic algorithm cannot have competitive ratio better than m for a certain online sequence of elements.

Exercise 0.5 The *minimum-weight perfect matching problem* is as follows: you are given a complete graph with positive edge weights, and the goal is to find a prefect matching with minimum total weight. This problem has a polynomial time algorithm. We will give a variant of the problem.

The *path-matching problem* is as follows: you are given a (1) graph with non-negative edge weights and (2) a set of terminals T (assume $|T|$ is even). The goal is to find $|T|/2$ edge-disjoint paths (i.e., paths that do not share an edge) in the graph with minimum total length such that ever terminal vertex is an end-point of exactly one of these paths.

Determine if the path-matching problem is in P or is NP-complete.