

Graded Homework 2

Lecturer: Mohsen Ghaffari

Student Name:

Note 1: Solutions must be typeset in \LaTeX and the related pdf file should be uploaded via moodle by 11:59 pm on December 8, 2021 (e.g., the pdf of your solution to problem i should be uploaded in the assignment titled “GHW2: Problem i ” on the moodle homepage). Late submissions will not be graded. If you would like to add a drawing in your solution, you can simply include a picture of a hand-drawn figure in your latex. You can submit only pdf one file for each problem and the drawing should be incorporated in the rest of your explanation.

Note 2: This is a theory course and any claim should be substantiated with a proof. You can discuss the problems with the other students but you should write your own solutions independently, and you should be able to orally explain your submitted solution to the instructors, if asked. It is strictly prohibited to share any (hand)written or electronic (partial) solutions with fellow students. Moreover, if you discussed a problem with another student, you must list their names on your submitted solution.

1 Randomized Ski Rental and Yao’s Principle (25 points)

Recall that in the lecture we demonstrated a deterministic 2-competitive algorithm for Ski Rental and noted that no $(2 - \varepsilon)$ -competitive deterministic algorithm exists for any $\varepsilon > 0$.

1. Show that, given a known input distribution (the algorithm knows in advance the probability of the season lasting exactly i days), there exists an at most 1.99-competitive deterministic algorithm. Hint: the standard algorithm shown in class is 1-competitive if the season length is of length at most B .
2. [Nothing to submit] Convince yourself that Yao’s principle is equivalent to proving “any randomized algorithm running on an unknown input distributions cannot be α -competitive” by proving “any deterministic algorithm running on a known input distribution cannot be α -competitive”. Conclude that, due to subtask 1, it is impossible to show a lower bound of 2 for competitiveness of randomized Ski Rental algorithms using Yao’s principle.
3. Yao’s principle is tight under very mild technical conditions. In particular, Yao’s principle is tight for Ski Rental. Demonstrate this by designing a 1.99-competitive randomized algorithm for Ski Rental (the algorithm does not know the probabilities in advance). Note: the competitive ratio in this subtask does not need to match nor reference the one in subtask 1). Hint: since the algorithm does not know the input distribution, the worst-case input can be assumed to be deterministic.

2 Maximizing Profit Minimizing Cost (25 points)

Updated on Dec 1. Deadline for problem 2 is extended to Dec 15.

We are given a stream of n elements $x_1, \dots, x_n \in \{1, \dots, n\}$, all distinct (i.e., the input is a permutation). We have to choose x_i, x_j with $j \geq i$ on arrival, meaning that when an element arrives, we must immediately decide if we include it in one of the two elements. The profit cost function is given as $x_j - x_i + 1 - \frac{1}{x_j - x_i + 1}$. We would like to minimize this cost. (Note that any sensible strategy will have $x_j - x_i + 1 - \frac{1}{x_j - x_i + 1} \geq 0$, as we can choose $i = j = n$).

1. Find a deterministic algorithm that achieves competitive ratio at most \sqrt{n} .
2. Show that any randomized algorithm has competitive ratio at least $\Omega(\sqrt{n})$

3 Near-Additive Spanners (25 points)

In class, we discussed construction of purely multiplicative and purely additive spanners. In this exercise, we will see a construction of $(1 + \epsilon, \beta)$ -spanners. These spanners are called *near-additive* spanners, as the multiplicative stretch is close to 1. These spanners are useful, as they give a stretch that is close to additive, but can be significantly sparser compared to purely additive spanners.

In the construction we have parameters d_1, d_2, p_1, p_2 , to be specified later. We have two sampled sets $S_1 \supseteq S_2$, where S_1 is obtained by sampling each vertex independently with probability p_1 , and S_2 is obtained by sampling each vertex from S_1 independently with probability p_2 . We say that a vertex $s \in S_1$ is s -heavy if it has at least d_2 vertices from S_1 in its $(\frac{1}{\epsilon} + 2)$ -neighborhood. Otherwise, we say it is s -light. We have the following guarantees:

1. Each vertex of degree at least d_1 , has a neighbor in S_1 with high probability.
2. Each s -heavy vertex in S_1 has a vertex from S_2 at its $(\frac{1}{\epsilon} + 2)$ -neighborhood with high probability.

We next assume that the above properties hold, which happens with high probability. We add the following edges to the spanner:

1. We add all edges adjacent to vertices of degree at most d_1 .
2. For each vertex of degree at least d_1 , we add an edge to a neighbor in S_1 .
3. We add BFS trees from all vertices in S_2 .
4. For each s -light vertex in S_1 we add the shortest paths from s to all vertices from S_1 in the $(\frac{1}{\epsilon} + 2)$ -neighborhood of s .

We next analyze the size and stretch of the spanner.

1. Choose values for d_1, d_2, p_1, p_2 such that the above properties hold with high probability, and such that the expected size of the spanner is minimized. You can assume that ϵ is a small constant. There is no need to optimize the logarithmic and constant terms in the size of spanner.
2. Prove that the spanner has stretch of $(1 + 4\epsilon, \Theta(1/\epsilon))$. You can assume that the properties discussed above hold, which happens with high probability. Guidance: divide the vertices into 3 types: light, medium and heavy, where light are vertices of degree at most d_1 , heavy are vertices of degree at least d_1 that have a neighbor in S_1 that is s -heavy, and medium are the rest. First analyze the stretch for pairs of vertices u, v , where the shortest $u - v$ path has a heavy vertex. For shortest paths that have only light and medium vertices, divide the path into segments of length at most $1/\epsilon$, and prove that in each of them we lose at most a $+4$ additive stretch.

4 Number of Minimum Cuts (25 points)

In the class, it was shown that the number of minimum cuts of any unweighted graph is at most $\binom{n}{2}$. Here we want to discuss two generalizations of this result.

1. In the lecture we proved that the number of minimum cuts in any unweighted graph is at most $\binom{n}{2}$. Prove the same result for *weighted graphs*. All the weights are positive.

2. For a graph $G = (V, E)$, a subset of edges $E' \subseteq E$ is a k -cut if $G' = (V, E \setminus E')$ has at least k connected components. Prove that the number of minimum k -cuts in any unweighted graph is at most $2^{O(k)} n^{2k-2}$.