Advanced Algorithms 2020 Deadline: ~~Dec 11~~, 2020, 11:59 pm
Extended to Dec 15

Graded Homework 2

Lecturer: Mohsen Ghaffari *Student Name:*

Note 1: *Solutions must be typeset in LaTeX and uploaded via moodle by 11:59 pm on Dec 11, 2020 (one pdf file for each problem). 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 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 Sparsest Cut (20 points)

(A) Give a deterministic polynomial sparsest cut algorithm for trees. That is, you are given a tree $T = (V, E)$ and your task is to output a set $S \subseteq V$ that minimizes the fraction

The edges of the tree are weighted.

$$\frac{E(S, V \setminus S)}{|S||V \setminus S|}.$$

E(S, V\S) is the summation of the weights of edges with exactly one endpoint in S.

(B) Give an $O(\log n)$-approximation randomized polynomial algorithm for sparsest cut in a graph G. Your algorithm should output such a cut with probability at least $1/2$. Your solution should use the result from part (A) and what you know about cut-preserving tree embeddings. You can assume the result from part (A) as a black-box even if you did not solve it.

2 Embedding for Trees (20 points)

In the class, we saw that every n-node weighted graph can be embedded into a $\text{poly}(\log n)$-dimensional space with L_1-distance with pairwise distance stretch at most an $O(\log n)$ factor. This exercise asks you to prove a similar fact, but for trees.

Prove that any n-node weighted tree $T = (V, E, w)$, where $w(e)$ denotes the length of edge e, can be embedded into a $k = O(\log n)$ dimensional space with L_∞-distance such that the distance between each pair of nodes remains exactly the same. More formally, prove that there is a mapping $f : V \to \mathbb{R}^k$ such that for every two nodes $u, v \in V$, we have $dist_T(u, v) = ||f(u) - f(v)||_\infty$. Recall that for two k-dimensional vectors $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_k)$, we have $||\mathbf{x} - \mathbf{y}||_\infty = \max_{i \in \{1,2,\ldots,k\}} |x_i - y_i|$.

Hint: In every N-node tree, there is a vertex v such that if we remove v, each connected component has at most $N/2$ vertices. If you use this hint, you do not need to prove it. Moreover, an embedding into a $\text{poly}(\log n)$ dimensional space with L_∞-distance would receive partial points.

3 Streaming 3-Connectivity (20 points)

In the class, we saw an algorithm with memory $n \cdot \text{poly}(\log n)$ that solves the connectivity problem in the streaming setting where we have a stream edge of arrivals and departures on a set V of n vertices. Devise an algorithm with $n \cdot \text{poly}(\log n)$ memory that solves the 3-connectivity problem in the same setting. That is, if the graph at the end of the stream is 3-edge-connected your algorithm should say YES with probability at least $1 - 1/n^2$, and otherwise it should say NO with probability at least $1 - 1/n^2$. Recall that a graph is 3-edge-connected if the graph remains connected after removing any set of at most 2 edges.

4 Communication with a Coordinator (20 points)

Consider n-players numbered $1, 2, \ldots, n$. A set $S \subseteq \{1, 2, \ldots, n\}$ of players is selected and revealed, where each player gets to know whether itself is in S or not, but it does not get to know about the other members. Each player can send a B-bit message to the coordinator, but unfortunately, the coordinate receives the bit-wise AND of the sent messages, instead of receiving each of them. Devise a scheme with $B = O(\log^2 n)$ so that the coordinator can still approximate $s = |S|$ up to a 2-factor, with probability $1 - 1/n^2$.

More formally, you should describe the B-bit message $m^i = (m^i_1, m^i_2, \ldots, m^i_B)$ that each player $i \in \{1, \ldots, n\}$ sends, as well as how the coordinator infers a 2-factor approximation $\tilde{s}$ of $s = |S|$ by receiving just the B-bit bitwise AND of these messages. That is, the coordinator will receive only $\big((m^1_1 \wedge m^2_1 \wedge m^3_1 \wedge \cdots \wedge m^n_1), (m^1_2 \wedge m^2_2 \wedge m^3_2 \wedge \cdots \wedge m^n_2), \ldots, (m^1_B \wedge m^2_B \wedge m^3_B \wedge \cdots \wedge m^n_B)\big)$. Your algorithm should guarantee that $\Pr[\tilde{s} \in [s, 2s)] \geq 1 - 1/n^2$.

5 Spanners via Ball Carving (20 points)

(A) Devise a deterministic ball carving algorithm that, given any unweighted undirected graph $G = (V, E)$, partitions vertices into disjoint clusters V_1, V_2, ..., such that: (1) The diameter of the subgraph $G[V_i]$ induced by each cluster is $O(\log n)$, (2) the set of intracluster edges $E' = \{e = \{u, v\} \in E | \exists i \text{ such that } v \in V_i \,\&\, u \in V_i\}$ has size $|E'| \geq |E|/2$. A randomized algorithm that achieves this guarantee with high probability will also be accepted.

(B) By choosing an appropriate subgraph in each cluster, and repeating the ball carving algorithm of part (A) on appropriately chosen graphs, devise an algorithm that computes a $O(\log n)$-multiplicative spanner with $O(n \log n)$ edges, in any n-node undirected unweighted graph. You can assume the result from part (A) as a black-box even if you did not solve it.