This problem set was last updated on Friday August 23, and has a total of five problems. Solutions should be handed in in written form before or after classes, and will be tentatively accepted until mid November.

Estimates of difficulties are indicated using *s:

1. (*) indicates a statement that follows fairly readily from the definitions, providing that they are understood thoroughly.

2. (**) usually require one of two steps of problem solving starting from the tools that we discussed.

3. (***) tends to require multiple steps, as well as additional tools that one can find on the internet.

Do keep in mind that these are only estimates: I highly recommended reading and thinking briefly about each problem.

Each problem is marked out of 5, and counts as much towards the overall grade. Collaboration is allowed/encouraged on problems, however each student must independently complete their own write-up, and list all collaborators. No credit will be given to solutions obtained verbatim from the Internet or other sources.

1. (**) Consider a random walk starting at vertex $u$ in an undirected graph with $m$ edges. Show that the expected time until the walk returns to $u$ is $\frac{2m}{\deg(u)}$.

2. (**) Show an algorithm that given a weighted tree and a value $d$, returns the number of pairs of vertices within (weighted) distance $d$ of each other in $O(n \log^2 n)$ time.

3. (**)
Finding a maximal $s$-$t$ min-cut: show that given a directed graph with two vertices $s$ and $t$, we can find a maximal min-cut between $s$ and $t$ in time proportional to the cost of computing a maximum flow between $s$ and $t$, plus $O(m)$ overhead.

Recall that a maximal min-cut is a min-cut $\hat{S}$ (with $s \in \hat{S}$) separating $s$ and $t$ such that for any other mincut $\tilde{S}$, we have $\tilde{S} \subseteq \hat{S}$. Your solution should also serve as proof that such maximal $s$-$t$ min-cuts always exist.

4. (***) Given a $O(n)$ time algorithm that takes as input a tree and a parameter $k$, and returns finding $k$ (possibly overlapping) paths (which can be succinctly represented by their end points) that cover the maximum number of vertices in the tree.
5. (**)

Another use of low stretch spanning trees is to embed the graph into a tree so that the weighted congestion of the tree edges is small. Here we’re given capacities $c_e$ and weights $w_e$ for all the edges. The weighted congestion of a tree edge $e$ is the total of

$$
\sum_{\hat{e} : e \in \mathcal{P}_T(\hat{e})} w_e \frac{c_e}{c_{\hat{e}}},
$$

where $e \in \mathcal{P}_T(\hat{e})$ denotes $e$ being on the shortest path in the tree $T$ between the endpoints of $\hat{e}$.

Note that there exists a tree such that if we embed the graph into it, the total weighted congestion is $O((\sum_e w_e + m)m^{0.01})$, and it can be found in $O(m^{1.01})$ time.

6. (*)

Consider the $\sqrt{n} \times \sqrt{n}$ grid with unit weighted edges. Show that for any parameter $k$, we can discard $O(n/k)$ edges so that the resulting graph has a low stretch spanning tree/forest with total stretch $O(n \log^3 k)$.

Note in particular this should imply that we can discard $O(n \log^{-10} n)$ edges from the grid so that the resulting graph has a tree with total stretch $O(n(\log \log n)^3)$.

7. (***)

A rainbow spanning tree in an undirected graph with colors on edges is a spanning tree where all the edges have different colors. Give an $O(n^{10})$ time algorithm for determining whether such a tree exists.

8. (*) Recall that the volume of a subset of vertices $S$ in an undirected graph is the total degrees of edges in it. A cut $S$ is balanced if its volume is in the range $[0.01m, m]$ (note that the total volume of all vertices is $2m$).

Show that for any parameter $\alpha$, a graph either has a balanced cut with conductance at most $\alpha$, or there is a subset $S'$ of volume at most $0.1m$ so that $G - S'$, the graph with $S'$ removed, has conductance at least $\alpha/10$. 

2