1. **Algorithmic engineering.** In this question, you are allowed to use the following tools we build during the class as black boxes:

- multiple-source shortest paths by Klein [4],
- cycle separators and r-division by Frederickson [2] and Klein-Mozes-Sommer [5],
- Monge heaps and FR-Dijkstra algorithm by Fakcharoenphol-Rao [1],

or any of their precedents. All other tools on planar graphs (even if you find them in the literature) have to be built from scratch. In particular, you are not allowed to use


as a black box. (But if you can rebuild their algorithm from scratch ...)

(a) Let $G$ be an $n$-vertex undirected planar graph with non-negative edge weights. Design and analyze an algorithm that computes a single-source shortest path tree in $O(n \log \log n)$ time.

*[Hint: Dijkstra algorithm runs in $O(n \log n)$ time on planar graphs out-of-the-box.]*

(b) Let $G$ be an $n$-vertex undirected planar graph, possibly with negative edge weights. Design and analyze an algorithm that computes a single-source shortest path tree in $O(n \text{polylog } n)$ time, or correctly reports that there is a negative cycle in $G$. What is the best running time you can get?

★(c) The difficulty in improving the running time in (b) seems to be that computing MSSP takes $\Theta(n \log n)$ time in general, and the recursion takes $O(\log n)$ levels. Can you improve the running time?

*[Hint: If you solve this problem, even with a $\log^* n$-factor over the state-of-art, you can publish a paper.]*

**References**


