This problem set has a total of 7 problems. You can do any 4 of them. Solutions can be either presented to the instructor or handed in.

1 Shortest Path in Polygons with Holes
The shortest path algorithm that we described in class only works for simple polygons. In more general cases, there may be obstacles in the middle of the polygon. Describe an $O(n^3)$ time algorithm for computing shortest paths in this more general case, where $n$ is the total number of vertices in the polygon and obstacles.

2 Shortest Cycle
Give an $O(m \log n)$ time algorithm for finding the shortest cycle involving a vertex $u$ in a directed graph with non-negative edge weights.

**NOT REQUIRED:** what about the shortest simple cycle involving $u$ in an undirected, unit weighted graph?

3 $r$-divisions
Recall that a $r$-division is a decomposition of a planar graph into $O(n/r)$ edge disjoint pieces such that each piece has:

1. $\leq r$ vertices, and
2. $O(\sqrt{r})$ boundary vertices.

Describe good $r$ divisions for:

1. a path on $n$ vertices, and
2. the wheel graph: a cycle on $n - 1$ vertices that are all connected to a center vertex, and
3. a $\sqrt{n} \times \sqrt{n}$ square grid.

4 Weighted Shortest Path with Two Distinct Positive Edge Weights
Give an $O(m)$ time algorithm for computing $s \to t$ shortest paths in a graph with only two distinct positive edge weights. It should take as input a directed graph where all edge weights are one of $w_1 > 0$ or $w_2 > 0$ along with vertices $s$ and $t$, and output a shortest $s \to t$ path in $O(m)$ time.

5 Counter Example to König’s Theorem on General Graphs
For bipartite graphs, König’s Theorem states that the size of the maximum matching equals to the size of the minimum vertex cover. Give a counter example to this for general graphs.
6 Maxflow on a Tree
Consider a variant of maximum flow on a tree: the (bi-directional) edges are capacitated, and vertices are either sources or sinks. That is, each vertex $u$ is either a source that can have up to $\text{out}(u)$ units of flow leaving, or a sink with up to $\text{in}(u)$ units of flow entering. Give an $O(n)$ time algorithm for routing the maximum amount of flow from sources to sinks.

7 Updating Flows Under Edge Removal
Give an $O(m \log n)$ time algorithm that takes a capacitated graph $G$ and a flow $f$ of value $F$, along with a set of edges $E^-$ with total capacity $C$, and outputs a flow $f^-$ with value at least $F - C$ that does not use edges in $E^-$. 