

# Problem Set 4

CS 331

Due Tuesday, October 24

1. You are given a sequence of  $n$  integers,  $x_1, \dots, x_n$ , and an integer  $k \in [n]$ . Each integer is polynomially large.

- (a) Give an  $O(n)$  time algorithm to find the contiguous subset of size at least  $k$  with maximum sum. That is, find two indices  $s, t \in [n]$  with  $t \geq s + k - 1$  that maximizes

$$\sum_{i=s}^t x_i.$$

Note that, since the integers may be negative, the answer may not be the whole sequence.

- (b) Now consider the contiguous subset of size at least  $k$  with maximum *average*:

$$\frac{1}{t - s + 1} \sum_{i=s}^t x_i.$$

For any threshold  $\tau$ , use part (a) to give an  $O(n)$  time algorithm to determine whether the maximum average is at least  $\tau$ .

- (c) Finally, use your algorithm in part (b) to give an  $O(n \log n)$  time algorithm to find the size at least  $k$  subset of maximum average.

2. Consider a weighted, directed graph where all distances lie in  $[1, 2)$ . We would like to find an  $O(m)$  time algorithm for single-source shortest paths on this graph.

- (a) Consider a variant of Dijkstra's algorithm that does not always visit the unvisited node of smallest  $c(u)$ , but instead arbitrarily picks one of the unvisited nodes of smallest  $\lfloor c(u) \rfloor$ . Show that such an algorithm still yields the correct answer.

- (b) Now give a data structure that allows this Dijkstra variant to run in  $O(m)$  time. **Hint:** at any point during the execution, the set of  $\lfloor c(u) \rfloor$  for unvisited nodes  $u$  can only have a small number of options.
- (c) Extend your result to  $O(mC)$  time and  $O(m)$  space for distances in  $[1, C)$  for any  $C \geq 1$ .
3. You are given a three dimensional object. On the horizontal plane it is an  $n \times n$  square, and on the vertical axis each square  $(x, y)$  is a square pillar rising to height  $h_{x,y} \geq 1$ . Adjacent pillars, even ones sharing corners, are fused together.

You submerge this object into a bucket of water, then carefully lift it out. Water will then drain off the sides, but it cannot drain through pillars. How many units of water will be captured in the object?

Since water flows downhill, it is only trapped up to the level at which there is a non-increasing path, horizontally and vertically, to an edge of the grid. For full credit, aim for an  $O(n^2 \log n)$  algorithm.

As an example, in the following grid 2 units will be captured, all in the center tile:

0	5	9
7	3	6
7	5	2

Any water in the center above level 5 will drain out to the north or south.

**Hint:** You may use from class that a variant of Dijkstra's algorithm can solve the *minimax path* problem. The shortest path problem is to find paths minimizing total length  $\sum_{e \in P} c(e)$ ; the minimax path problem is to find paths minimizing the *maximum* length  $\max_{e \in P} c(e)$ .