CS 598TMC, Fall 2019
Timothy Chan

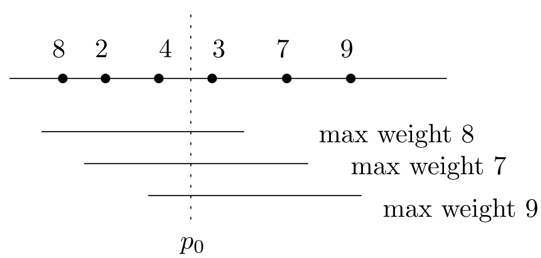
Homework 4 (due Dec 4 Wednesday (11am in class))

Instructions: You may work in groups of at most 2. Hand in one set of solutions per group. Acknowledge any discussions you have with other students and other sources you have consulted. Solutions must be written *in your own words.*

1. [*30 pts*] We are given a set S of n points in 1D, where each point p has a weight $w(p)$, and an integer $k \leq n$. We want to find an interval I^* containing exactly k points whose largest weight is minimized.

   (a) [*15 pts*] First consider the special case of the problem where I^* is known to contain a fixed point p_0. Describe a dynamic data structure for this problem that supports insertions and deletions of points in S (and maintains I^* after each update) in $O(\log^c n)$ time (or better) for some constant c.

   [Note: You may use known dynamic data structures for 1D *range max* queries (finding the largest weight among points in a query interval), which have $O(\log n)$ time for insertions, deletions, and queries.] [Hint: binary search. . .]

   for $k = 4$, answer is 7

   (b) [*15 pts*] Now, using (a), describe a dynamic data structure for the general problem that supports insertions and deletions of points in S and finds I^* in $O(\log^c n)$ (amortized) time.

2. [*30 pts*] We are given a set S of n points in a constant dimension, and an integer $k \leq n$. Consider the problem of finding a ball b^* that contains at least k points of S with the minimum radius.

   (a) [*15 pts*] Describe an efficient algorithm with a constant approximation factor for this problem. Aim for $O(n \log n)$ running time or better.

   [Note: weaker results may still receive partial credit, e.g., $O(n \log U)$ or $O(n \log n + kn)$. . .] [Hint: use Z-ordering and shifting.]

   (b) [*15 pts*] Describe an efficient dynamic data structure to find a constant-factor approximation to b^* under insertions and deletions of points in S. Aim for $O(\log^c n)$ (amortized) update time for some constant c.

[Note: weaker results may still receive partial credit, e.g., $O(\log^c U)$ or $O(k \log^c n)\ldots$] [Hint: the result from Problem 1 may be useful.] [Bonus/research question: dynamic $(1+\varepsilon)$-factor approximation?]