E0 225: Homework 1

Deadline: Oct 14th, 5pm

Instructions

- All problems carry equal weight.
- You may discuss these problems with others. However, you must *write* your own solutions and list your collaborators for each problem.
- Academic dishonesty/plagiarism will be dealt with severe punishment.
- Late submissions are accepted only with prior approval or medical certificate.

1 (Skyline points) Let P be a set of n points in the plane. A point $p(p_x, p_y)$ *dominates* another point $q(q_x, q_y)$ if and only if $p_x > p_y$ and $q_x > q_y$. Any point in P which is not dominated by any other point in P is a *skyline* point. The goal here is to compute all the skyline points of P. As a motivating example, think of each cricketer as a point in the plane, and the x-axis (resp., y-axis) to be the runs scored (resp., wickets taken). Computing skyline points is a popular approach to capture a potential set of good points (in this case, all-rounders in cricket).

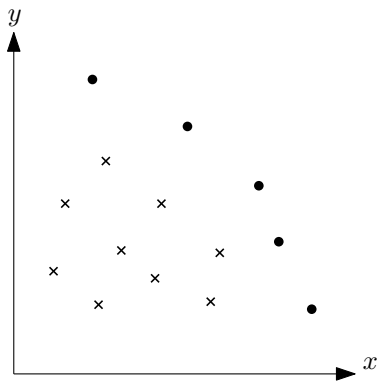

Figure 1: The skyline points are the black disks and the dominated points are the crosses.

Here is an algorithm to compute the skyline points.

skyline(P):
1. if $|P| = 1$, then return P.
2. divide P into the left and right halves P_ℓ and P_r by the median x-coordinate.
3. *discover* the point p with the maximum y-coordinate in P_r.
4. *prune* all points from P_ℓ that are dominated by p.
5. return the concatenation of skyline(P_ℓ) and skyline(P_r).

Prove that the above algorithm runs in $O(n \log h)$ time, where h is the number of skyline points in P. (*Hint: Define $T(n, h)$ to be the running time of the algorithm on an input of size n and output size h. Write a recurrence in terms of $T(n, h)$ and establish an important base case of $T(n, 1) \leq cn$, where c is a constant independent of n.*)

2 (Alternate median finding algorithm)

1. Let $T(n) = \sum_{i=1}^{k} T(a_i n) + O(n)$, where $a_i > 0$ for $i = 1, \ldots, k$ and $\sum_{i=1}^{k} a_i \leq 1$. Use the substitution method to prove that

$$T(n) = \begin{cases} O(n), & \text{if } \sum_{i=1}^{k} a_i < 1 \\ O(n \log n), & \text{if } \sum_{i=1}^{k} a_i = 1. \end{cases}$$

2. In the median finding algorithm described in the lecture, the input elements are divided into groups of 5. The goal of this exercise is to understand if there is anything special about this number 5. Will the algorithm work in linear time if they are divided into groups of 9? If yes, then prove it. Will the algorithm work in linear time if they are divided into groups of size 3? Make use of the recurrence proven above.

Practice problems (will not be graded)

1. Consider the recurrence $T(n) = 4T(n/3) + n$. Show that a substitution proof with the assumption $T(n) \leq cn^{\log_3 4}$ fails. Then show how to subtract off a lower-order term from $cn^{\log_3 4}$ to make the substitution proof work. (*This exercise is to illustrate one of the subtlety while using the substitution method*).

2. You might want to think of a divide and conquer algorithm which computes the skyline points of P in $O(n \log n)$ time.