CS 6301.008 Homework 1

Due Tuesday February 6th at 11:30am, on eLearning

January 23, 2018

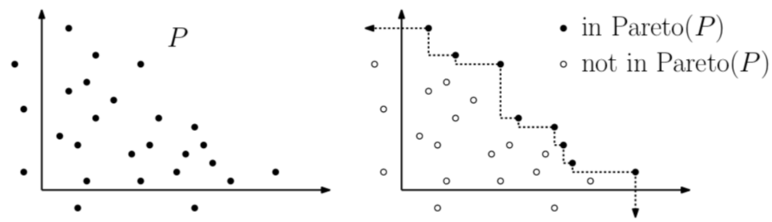
Please answer all three questions. You may form groups of up to three students. Each group should write a single set of solutions with each group member's name and Net ID on the front page. Each group member should then submit a copy through eLearning.

Important Course Policies and Advice

- Generally, when asked to "design and analyze an algorithm", you should explain the algorithm clearly in English and/or with pseudocode and explain its running time.
- With all problems, you need to provide a justification/informal proof that your algorithm/answer is correct. Remember, the goal is to explain *why* you are correct and not simply rewrite your algorithm/answer again.
- You may use anything from the lecture, the required textbook or lecture notes, or prerequisite courses without further explanation. If you want to give an algorithm that differs slightly from something in class, feel free to just explain the differences from what we did, along with an explanation for why you're using those differences. Simple (correct) solutions that merely perform a reduction to something we did in class are *highly* encouraged. Don't try to reinvent the wheel!
- You may assume that reasonable operations involving a constant number of geometric operations may be done in $O(1)$ time. You may also assume inputs lie in general position (no two points share an x-coordinate, no three points lie on a line, no four points lie on a circle, etc.). Clearly state your assumptions if they are not something we already used in lecture.
- I strongly suggest you use LaTeX to typeset your solutions. Any illegible solutions will be considered incorrect. The announcement for this homework links to a template for writing solutions in LaTeX.
- If you use outside sources or figure out solutions in collaboration with anybody outside your group, then you may cite that source or person and still receive full credit for the solution. Material from the lecture, the required textbook, or prerequisite courses need not be cited. Failure to cite other sources or failure to provide solutions in your own words, even if quoting a source, is considered an act of academic dishonesty.

See https://utdallas.edu/~kyle.fox/courses/cs6301.008.18s/ for more detailed policies and some advice. If you have any questions about these policies, please do not hesitate to ask in class, in office hours, or through email.

1. Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in the plane ($\mathbb{R}^2$), and let $p_i = (x_i, y_i)$ for each i. The *Pareto set* $\text{Pareto}(P) \subseteq P$ is the subset of P containing each point p_i such that there exists no point $p_j \in P$ with $i \neq j$ such that both $x_j \geq x_i$ and $y_j \geq y_i$. In other words, each point of $\text{Pareto}(P)$ has no other point of P both above and to the right of it; you can visualize them as the turns on a staircase:



The set $\text{Pareto}(P)$ (from Mount).

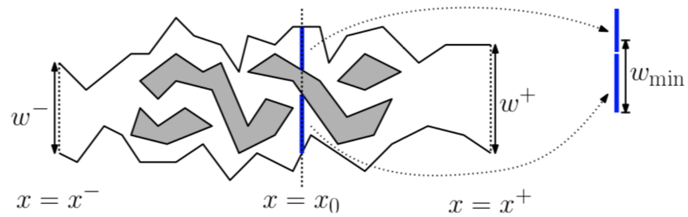
(a) Describe and analyze an $O(n \log n)$ time algorithm to compute/output $\text{Pareto}(P)$. *[Hint: You might consider writing a modification of Graham's scan.]*

(b) Let $h = |\text{Pareto}(P)|$. Describe and analyze an $O(nh)$ time algorithm to compute/output $\text{Pareto}(P)$. *[Hint: You might consider writing a modification of Jarvis's march.]*

(c) Describe and analyze an $O(n \log h)$ time algorithm to compute/output $\text{Pareto}(P)$. For simplicity, you may assume that the value h is known in advance. *[Hint: You might consider writing a modification of Chan's convex hull algorithm.]*

2. Let $C = \{c_1, \ldots, c_n\}$ be a set of n circles in $\mathbb{R}^2$ where each circle c_i is given as its center point $a_i = (x_i, y_i)$ and radius $r_i > 0$. Note that some circles may be completely nested in one-another without intersecting. Describe and analyze an $O(n \log n)$ time algorithm to determine whether or not any pair of circles intersect. *[Hint: Describe a plane sweep algorithm. To describe one, it suffices to explain 1) what is stored for the sweep-line status and what data structure is used, 2) what are the event points and what data structure is used for the event queue, and 3) how to check for intersections and update the sweep-line status and event queue during events. You should also explain* why *handling events as you do is correct.]*

3. Let P be an x-monotone polygon that is bounded between two vertical lines $x = x^-$ and $x = x^+$ (see the figure). Inside P are some number of disjoint x-monotone polygons. Let n denote the total number of vertices on P and the other polygons.



The setting (from Mount). Ignore the additional notation I did not define.

Describe and analyze an $O(n \log n)$ time algorithm that returns the length of the shortest vertical line segment intersecting either two non-adjacent edges of P or both an edge of P and an edge of one of the internal polygons.