CS 506, HW1

Prof. Jared Saia, University of New Mexico

Due: Feb. 18th

You are encouraged to work on the homework in groups of 2 to 4 people. You may turn in one writeup per group, but please certify that all members worked on each problem. Note that some of these problems are from the book "Computational Geometry (third edition)" by Berg, et al. Mount's notes and homework problems are available in the link off the course web page: http://www.cs.umd.edu/class/fall2016/cmsc754/Handouts/cmsc754-2016-08-handouts.pdf

1. (Exercise 8.1 from Berg) Prove that the duality transform discussed in class is indeed incidence and order reversing.

2. (Exercise 8.2 from Berg) The dual of a line segment is a left-right double wedge, as discussed in class. Answer the following

   (a) What is the dual of the collection of points inside a given triangle with vertices p,q and r

   (b) What type of object in the primal plane would dualize to a top-bottom double wedge?

   Solution: (a) Completely Correct: It is the union of the three left-right double wedges that are the duals of the line segments pq, qr and rp. To see this, note first that the boundary of the triangle dualizes to these three wedges. What about points inside the boundary? Note that order preservation says that if all points, x, in the triangle are above say line pr and below lines pq and qr, then in the dual plane, x* will be below point pr* and above pq* and qr*. So x will be inside one of the wedges. (b) Completely correct. Consider a horizontal line that is missing all points in the line segment with x values (-x',+x'). This line dualizes to a top-bottom wedge. To see this, note that the right part of this object dualizes to lines with slopes x' to up to infinity, and the left part of the object dualizes to lines with slopes -x' down to -infinity.

3. In the online convex hull problem, we are given a set of n points one at a time. After receiving each point, we compute the convex hull of all points seen so far. Consider this problem in the 2D plane. Give an efficient online algorithm to update the convex hull when a new point is given. Analyze your algorithm.

   Solution: Implicit binary search to find the tangent lines. Need to also say how you can "snip out" all the points in the prior convex hull between the tangent vertices in logarithmic time. For example, if you store the hull points in a balanced tree like a skip list, you can delete any number of points between some keys x and y in constant time via pointer operations.

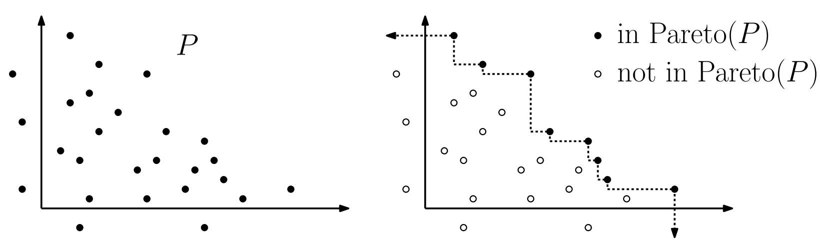


Figure 1. Example Pareto optimal figure (from David Mount hw).

4. Problem 2, HW 1 from David Mount's class (Pareto Optimal/Convex Hull problem) quoted below.
   Consider a set of points $P = \{p_1, \ldots p_n\}$ in the plane where $p_i = (x_i, y_i)$. A *Pareto set* for P, denoted $Pareto(P)$ is a subset of points P' such that for each $p_i \in P'$, there is no $p_j \in P$ such that $x_j \geq x_i$ and $y_j \geq y_i$. That is, each point of $Pareto(P)$ has the property that there is no point of P that is both to the right and above it. Pareto sets are important whenever you want to optimize *two* criteria (e.g. accuracy and precision of a machine learning algorithm, cheapness and flight "shortness" for airline tickets, etc.), since they represent the optimal "envelope" of possible solutions.
   This problem explores the many similarities between Pareto sets and convex hulls. Whenever a problem asks for an algorithm, briefly justify correctness of your algorithm, explain any non-standard data structures, and derive the runtime.

(a) A point p lies on the convex hull of a set P if and only if there is a line passing through p such that all the points of P lie on one side of this line. Provide an analogous assertion for the points of $Pareto(P)$ in terms of a different shape.

(b) Devise an analogue of Graham's convex-hull algorithm for computing $Pareto(P)$ in $O(n \log n)$ time. Briefly justify your algorithm's correctness and derive its runtime. (You don't need to explain the algorithm "from scratch"; you can explain what modifications need to be made to Graham's algorithm.

(c) Devise an analogue of Jarvis march algorithm for computing $Paret(P)$ in $O(hn)$ time where h is the cardinality of $Pareto(P)$. (as in the last part, you can just explain the differences with Jarvis's algorithm.

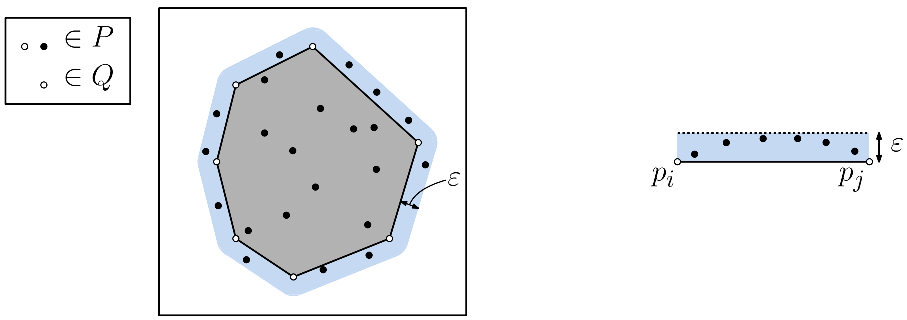
(d) Devise an algorithm for computing $Pareto(P)$ in $O(n \log h)$ time. Hint: Chan!

Solution: (a) Correct: For point (a, b) to be in the Pareto(P), all points in P must be either be below the line $y = b$ or to the left of the line x = a. (b) Sort points in descending order by x-coordinate. Add rightmost point to Pareto(P). Process points in decreasing order of x-coordinate and if we get a point with larger y-coordinate add it to Pareto(P). So we just check y-coordinate values of new points against last y-coord in Pareto(P), instead of checking for CCW turns. Runtime is $O(n \log n)$. (c) We want to start with top-most point and look for next highest point with greater x-coordinate. So each time we add something to the Pareto(P), we do n possible checks. This gives $O(nh)$ runtime.(d) Initially pretend we know h. Break points into sets of size h and find the mini-Paretos of each partition in time O(h log h) using Grahams. Next, merge by starting with the rightmost point of all sets. Say p is the most rightmost point. Put it in the Pareto. Now look at the rightmost points of the remaining sets and if they are below p, pop them off. Now find the next rightmost point and continue. Takes $O(n \log h)$ time to merge. Finally, we need to "guess" h iteratively as in class in the case where we don't know it initially.

5. Let R be a set of n red points in the plane, and let B be a set of n blue points in the plane. A **separator** or R and B is a line ℓ that has all points of R to one side and all points of B to the other. Give an algorithm that can decide in $O(n^2)$ time whether R and B have a separator.

6. (Exercise 8.10 from Berg) Let L be a set of n non-vertical lines in the plane. Suppose that the arrangement $\mathcal{A}(L)$ only has vertices with level 1. What can you say about this arrangement? Next suppose that lines of L can be vertical. What can you say now about the arrangement?

   Solution: (a) Correct: If no lines are vertical, and all vertices are at level 1, then there can only be vertex in the arrangement. Easy to show this via a proof by contradiction. (b) Correct: If the lines of L can be vertical, then we can have n-1 vertical lines, and one non-vertical line which will create n-1 level 1 vertices.



(a) (b)

Figure 2. Computing an ϵ-sketch (from David Mount hw)

7. Problem 4, HW 3 from David Mount's class (ϵ sketch of convex hull), quoted below.
   You are given a set P of n points lying in the unit square in the plane. Given any subset $Q \subseteq P$, clearly we have $\text{conv}(Q) \subseteq \text{conv}(P)$. For $\epsilon > 0$, we say that Q is an ϵ-sketch of P if every point of P lies within distance at most ϵ of $\text{conv}(Q)$ (See Figure 2 (a)).

   (a) Consider the following simple greedy algorithm for computing an ϵ-sketch of a planar point set P. First let $\langle p_0, \ldots, p_{k-1} \rangle$ denote the vertices of $\text{conv}(P)$ listed in counterclockwise order. Put p_0 in Q and set $i \leftarrow 0$. Find the largest index j, $i < j \leq k$ such that all the points $\{p_{i+1}, \ldots p_{j-1}$ lie within distance ϵ of the line segment p_i, p_j (See Fig 2(b)). (Indices are taken modulo k so $p_k = p_0$.) If $j = k$ then stop. Otherwise, add p_j to Q, set $i \leftarrow j$ and repeat. Show that this procedure correctly produces an ϵ-sketch of P.

(b) Fix some $\epsilon > 0$. Let $m(P)$ be the minimum number of points on any ϵ-sketch of P. Assume the points of P are in convex position, i.e. they all lie on the border of conv(P). Show that the greedy procedure above produces a sketch of size at most $m(P) + 1$.
Hint: First show that the points in any minimum sketch can be assumed to be taken from the vertices of the convex hull of P. To do this, show that you can take any arbitrary ϵ-sketch and create an ϵ-sketch where all vertices are in P, by only increasing the number of vertices by a constant factor. Second, consider the points of the minimum sketch in cyclic order about P's convex hull and ask how many points in the original hull can occur between any two points of the minimum sketch.

(c) Prove that $m(P) = O(1/\epsilon)$. I.e., that it it is completely independent of the number of vertices of the convex hull. What implication does this have on the amount of space needed to store an ϵ-approximation of all crepe recipes?

Solution: (a) Correct: Showing that the greedy algorithm produces a ϵ sketch is just a direct proof; (b) Correct. We can ensure all vertices are in P by increasing the number of vertices in the sketch by at most a constant factor. To see this, consider two adjacent vertices p_i and p_j in the initial sketch and rotate the plane so that the edge between them is horizontal. Consider all points, S (in P) that are at most ϵ above this line segment, and sort the points in S by x coordinate. Note that the y coordinates of all these points differ by at most 2ϵ. So we can cover all points in S by including the first vertex in S, and 4 additional vertices to ensure that there is never a vertical distance of more than ϵ between adjacent vertices. Continuing in this way, we can make all vertices points in P, while blowing up the number of vertices by at most a factor of 4. (b) Correct. To show that greedy takes m(P) + 1 points, consider some optimal ϵ-sketch OPT. Note that there's at least 1 vertex in OPT between any two vertices selected by greedy, say v and v'. If OPT does not contain at least one vertex in the sequence between v and v', then there is some point that is not within ϵ of the line segment chosen by OPT that is closest to these points. This holds by the definition of the greedy algorithm. (c) Correct: the perimeter of any convex shape inside the unit square is no more than the perimeter of the unit square. Hence, the perimeter of the convex hull is $O(1)$. Thus, in the worst case, $O(1/\epsilon)$ points suffice to ensure that every point on the perimeter is within ϵ of the sketch.

8. Challenge: (This is the type of problem that could turn into a project or potentially a paper) Can you adapt the ϵ-sketch convex hull problem to come up with a similar type of sketch of the upper envelope in an arrangement? What can you say formally about the number of lines in your sketch of the upper envelope and how well the sketch approximates the true upper envelope? (Super Challenge: Any connections to sketching a Voronoi diagram?)

   Solution: Correct. A precise approximation guarantee. OK to turn in this problem in the next homework. Connection between upper envelope problem and convex hull pointed out.