

Due: 23rd of March 2011 at 3pm.

COMP 5045 – Assignment 1

As a first step go to the second page and read the section: Advice on how to do the home work.

1. Let \mathcal{S} be a set of n points in the plane sorted with respect to their y -coordinate from bottom to top. Give an algorithm that computes the convex hull of \mathcal{S} in $O(n)$ time. Argue its time complexity. [10 points]
2. Design an algorithm that given a simple polygon P (the vertices of P are given in counter-clockwise order) in the plane and a query point q decides if the point is inside or outside the polygon. The faster the better. Prove the correctness of the algorithm and the running time of the algorithm. Can your algorithm be extended to the case when P is a polygon with holes? Argue why or why not. [10 points]
3. In many situations we need to compute convex hulls of objects other than points. For instance, let \mathcal{S} be a set of n hexagons in the plane. A hexagon is a simple, not necessarily convex, polygon with six vertices. Explain how to compute the convex hull of \mathcal{S} in time $O(n \log n)$ and prove correctness of your algorithm. [10 points]
4. Let \mathcal{S} be a set of n disjoint line segments in the plane, and let p be a point not on any line segment of \mathcal{S} . Determine all line segments of \mathcal{S} that p can see, that is, all line segments of \mathcal{S} that contain some point q so that the open segment pq does not intersect any line segment of \mathcal{S} . Give an $O(n \log n)$ time algorithm. [10 points]
Hint: Try to rotate (“sweep”) a half-line around p .
5. In the lecture we showed that every simple polygon with more than 3 vertices has a diagonal. Construct an algorithm that given a simple polygon \mathcal{P} with $n > 3$ vertices, computes in $O(n \log n)$ time a diagonal that splits \mathcal{P} into two simple polygons \mathcal{P}_1 and \mathcal{P}_2 with at least $\lfloor n/3 \rfloor - 2$ vertices each. Prove the correctness of your algorithm. [10 points]
Hint: Consider the dual graph of a triangulation.

Advice on how to do the home work

- Be careful with giving multiple or alternative answers. If you give multiple answers, then we will give you marks only for "your worst answer", as this indicates how well you understood the question.
- Some of the questions are very easy (with the help of the lecture notes or book). You can use the material presented in the lecture or book (without proving it). You do not need to write more than necessary (see comment above).
- When giving answers to questions, we always would like you to prove/explain/motivate your answers.
- When giving an algorithm as an answer, the algorithms does not have to be given as (pseudo-)code.
- If you do give (pseudo-)code, then you also have to explain your code and your ideas.
- Unless otherwise stated, we always ask about worst-case analysis, worst case running times etc. (for example, hashing does not guarantee anything in the worst case, in general).
- As done in the lecture, and as it is typical for an algorithms course, we are interested in the most efficient algorithms. Hence, the faster your algorithm, the more points we can give you.
- It might help you (and us) if you briefly describe your general idea, and after that you might want to develop and elaborate on details. If we don't see/understand your general idea, we cannot give you points for it.
- If you use further resources (books, scientific papers, the internet,...) to formulate your answers, then add references to your sources.