Data structures project, Definitions document

Heikki Haapala and Aleksi Markkanen
Student numbers 014090190 and 013126382
6 pages

January 17, 2013

Contents

1 Introduction

The topic of choice for the project is the problem of finding the convex hull for a set of points. This is a problem in the field of computational geometry. The convex hull of a set of points can be defined thusly. Let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a finite set of points that lie on a plane. Then we say that the convex hull of the set S is the smallest convex set C that satisfies that $S \subset C$. It is easy to see that if $S' \subset S$ then for the convex hulls C and C', respectively, it holds that $C' \subset C$.

Since S is finite, the set C is a convex polygon. Thus, we know C uniquely from the set of its vertices.

We will compare the performance of several algorithms in different situations.

2 Chosen algorithms and data structures

We will implement three different algorithms: *Gift-wrapping algorithm*, *Graham scan* and *Quickhull*. We also implement the *Akl-Toussaint heuristics* to discard large amount of points as a preprosessing step.

We will also need a sorting algorithm since the *Graham scan* supposes that the input is ordered. Points will be stored in a linked list.

3 Inputs

Our program will take as input a list of points for which the convex hull needs to be computed.

When using the *Akl-Toussaint heuristics* we will choose from the inputted points those with the largest and smallest x and y coordinates. We will also pick points that have the minimum and maximum sums and differences of x and y coordinates. We can safely discard all the points that lie in the thusly formed octagon.

After the preprosessing we will input the points to the algorithms of choice.

4 Computational complexity

Gift-wrapping algorithm is an output sensitive algorithm. Its time complexity depends on its output. The time complexity is $\mathcal{O}(hn)$ where h is the amount of vertices of the convex hull. As the amount of vertices can only depend linearly on the input length, the worst performance is $\Theta(n^2)$.

Graham scan has the time complexity of $\mathcal{O}(n)$ for ordered input. We will sort the input using *Mergesort* that has the time complexity of $\mathcal{O}(n \log n)$. This brings the total complexity to $\mathcal{O}(n \log n)$.

Quickhull has the average case complexity of $\mathcal{O}(n \log n)$ but it can degenerate to $\mathcal{O}(n^2)$ in the worst case.

Space complexity is in every case $\mathcal{O}(1)$.

References

[1] Convex hull algorithms,
Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Convex_hull_algorithms