Design and Analysis of Algorithms: Lecture 2

Ben Chaplin

Contents

1 Divide & Conquer

1.1 Paradigm

Given a problem of size n, the general strategy of divide & conquer can be summarized:

1. Divide the problem into a ($a \geq 1$) subproblems of size $\frac{n}{b}$ ($b > 1$).
2. Recursively solve subproblems.
3. Combine solutions of the subproblems into one overall solution.

2 The Convex Hull problem

2.1 Definitions

Definition. Given a set of points S on the real plane, **convex hull of S** ($CH(S)$) is the smallest convex polygon containing all of the points in S.

We'll represent a convex hull as a sequence of points in clockwise order.

2.2 Brute-force algorithm

Algorithm 1 Brute-force algorithm for the Convex Hull problem

Input

S a set of points in the real plane, where:

1. no two points have the same x-coordinate,
2. no two points have the same y-coordinate,
3. and no three points are colinear

Output

$CH(S)$ the convex hull of S (as a sequence of points in clockwise order)

```
1: procedure CONVEXHULL(S)
2:     C ← {}
3:     for each pair of points p, q ∈ S do
4:         Draw the line which holds both points
5:         if all remaining points lie on one side of the line then
6:             Add the points p, q to C
7:         end if
8:     end for
9:     return the elements of C, clockwise-ordered
10: end procedure
```

2.3 Brute-force runtime

The brute-force algorithm:

1. iterates $O(n^2)$ times (line 2)
2. during each, iterates $O(n)$ times (line 4)

The sorting of the elements before returning runs in linear time and occurs only once, therefore, **Algorithm 1** runs in $O(n^3)$ time.

2.4 Divide & conquer algorithm

Definition. Given a vertical line and two points on either side of it p, q, define $y(p, q)$ to be the y-intercept of the point where the vertical line intersects the line containing p and q.

Algorithm 2 Divide & conquer algorithm for the Convex Hull problem

Input

S a set of points in the real plane, where:

1. no two points have the same x-coordinate,
2. no two points have the same y-coordinate,
3. and no three points are colinear

Output

$CH(S)$ the convex hull of S (as a sequence of points in clockwise order)

```
1: procedure DIVIDE(S)
2:     m ← the median x value for all the points in S
3:     if |S| > C then                          ▷ any constant C will work
4:         A ← {(x, y) ∈ S | x ≤ m}
5:         B ← {(x, y) ∈ S | x > m}
6:         CH(A) ← DIVIDE(A)
7:         CH(B) ← DIVIDE(B)
8:     else
9:         CH(A) ← CONVEXHULL(A)                ▷ from Algorithm 1
10:        CH(B) ← CONVEXHULL(B)
11:        return MERGE(CH(A), CH(B), m)
12:    end if
13: end procedure
```

procedure MERGE(A, B, m)
    $i, j \leftarrow 1$
    $a_i \leftarrow i$-th point in A, ordered counterclockwise, from the point with the $x-$value closest to m
    $b_j \leftarrow j$-th point in B, ordered clockwise, from the point with the $x-$value closest to m
    while $y(a_{i+1}, b_j) > y(a_i, b_j)$ or $y(a_i, b_{j+1}) > y(a_i, b_j)$ **do**
        if $y(a_{i+1}, b_j) > y(a_i, b_j)$ **then**
            i++
        else
            j++
        end if
    end while
    $u \leftarrow (a_i, b_j)$ ▷ the upper tangent
    (repeat the above procedure, but order A clockwise and B counterclockwise)
    $l \leftarrow (a_{i'}, b_{j'})$ ▷ the lower tangent
    return $CH(A \cup B)$:
    (being the segments in A until a_i, u, the segments in B starting at $a_{i'}$ up until $b_{j'}$, and l)
end procedure

2.5 Divide & conquer runtime

The divide and conquer algorithm:

1. divides the problem in half $O(log(n))$ times
    (a) for each divide, we must find a median, which takes $O(n)$ time
2. finds a convex hull in $O(1)$ time (as we've divided down to a constant factor)
3. merges two convex hulls $O(log(n))$ times
    (a) each merge takes $O(n)$ time

Therefore, **Algorithm 2** runs in $O(nlog(n)) + O(nlog(n)) = O(nlog(n))$ time.