Emo Welzl, Mittagsseminar, 16. Mai 1995

Two Exercises on Convex Hulls

Given n points in the plane, their convex hull can be computed in time $O(n \log h)$, if h is the number of vertices of the convex hull; upper and matching lower bound have been proved by Kirkpatrick & Seidel. Anyway, we don't want to tackle that complicated problems today, and start with something much easier.

EXERCISE 1. *Given k convex polygons $P_1, \ldots P_k$ in the plane (not necessarily disjoint) with overall n vertices, compute their convex hull C in time $O(kh + n)$, h the number of vertices of the convex hull.*

In order to solve the exercise, we proceed similar to Jarvis wrap, and we borrow some of the analysis ideas from Graham scan. First, we determine for each polygon P_i the lexicographically smallest vertex s_i, and we assume that s_1 is the lexicographically smallest point among those vertices. Clearly, $c_1 = s_1$ constitutes also a vertex of the convex hull C we want to compute. Next we determine the upper tangent from c_1 to each of the polygons P_i, $i = 1, \ldots k$; let s_i' be the contact point on P_i on this tangent (if the contact occurs along an edge e of P_i, then take the vertex furthest away from c_1 on this edge e).

Since c_1 has been chosen in such a way that it cannot lie in the interior of P_i (for any i), the tangent exists, and we can find this tangent by starting from s_i and proceeding in P_i in clockwise order until the tangent is found. Note that all vertices[1] we pass in this process in P_i disqualify from being potential vertices of C and thus can be removed from P_i. The "topmost" of these tangents is supporting line of the edge incident to c_1 in clockwise order, and the corresponding contact point $c_2 = s_j'$ is the next vertex in C. (It is left to the experienced degeneracy handling reader to resolve possible ambiguities from now on).

Now we determine tangents from c_2 to each of the P_i's. To be precise, we mean the directed tangent from c_2 which has P_i on its right side. Again we can search for such a tangent starting from s_i', eliminating vertices encountered on the way. We proceed in this way to the next vertex c_3 on C, and so on, until we reach $c_{h+1} = c_1$.

How much time have we spent. Initializiation takes time $O(n)$. For each c_j, we have to spend time $O(k + w_j)$, w_j the number of points we meet in the tangent searching process. Every such vertex is eliminated after encounter, so the w_j's accumulate to a number less than n. Hence, we get an overall bound of $O(kh + n)$, as suggested by the exercise.

The next exercise assumes that some oracle preinforms us about the number of vertices of the convex hull of a finite point set, and asks for a fast output sensitive procedure in that case.

Exercise 2. *Assuming knowledge of the number g of the number of vertices of a given set S on n points in the plane, compute the convex hull C of S in time $O(n \log g)$.*

That's really easy, at least after having settled Exercise 1. First partition S into $k = \lceil n/g \rceil$ sets $S_1, \ldots S_k$ of size at most g each. Compute the convex hull C_i of S_i, for each $i = 1, \ldots k$. This takes time $k\, O(g \log g) = O(n \log g)$ altogether, if we use Graham scan, say. Now apply Exercise 1 to the resulting convex polygons. Done.

We got dangerously close to an $O(n \log h)$ solution, if we only knew h in advance. Here a standard trick – exponential search – fails, but superexponential search will do:

For $g = 2^{2^t}$, $t = 0, 1, \ldots$, employ the algorithm from Exercise 2, but – for each g – stop and proceed to the next g, if the number of steps performed exceeds $cn \log g$, (where c is a constant which guarantees successful termination if $g \geq h$). The overall running time is

$$O(n \sum_{t=0}^{\ell} \log(2^{2^t})) = O(n \sum_{t=0}^{\ell} 2^t) = O(n 2^{\ell+1}) \ ,$$

with ℓ the smallest integer with $2^{2^\ell} \geq h$. That is, $2^{\ell-1} < \log h$ and so $2^{\ell+1} = O(\log h)$.

This new and elegant $O(n \log h)$ algorithm is due to Timothy Chan, and it generalizes to a simple optimal $O(n \log h)$ algorithm for convex hulls in 3-space, using Dobkin-Kirkpatrick hierarchies for the phase of the algorithm described in Exercise 1. Here it is important to observe, that Exercise 1 overfulfills the requirements of Exercise 2, since an $O(kh \log h + n)$ would still suffice. Previous optimal output-sensitive solutions in 3-space were either randomized (Clarkson & Shor), or algorithms derived from that randomized procedure by involved (and presumably impractical) derandomization techniques (Chazelle & Matoušek).

[1] Some care is in place – the first vertex s_i may still be on the convex hull.