

Homework 7  
Fundamental Algorithms, Fall 2001, Professor Yap

- DUE: This homework is not graded but will help you prepare for the latest topics in the final exam.
- For maximum effectiveness, I suggest that you try to solve the problems yourself before looking at the answers.

- 
1. Problem 33.1-4, page 939. Determine in  $O(n^2 \log n)$  time whether a set of  $n$  points is “degenerate” in the sense that 3 of the points are collinear.
  2. Verifying Extremal Point Set. Suppose you are given a set  $S$  of  $n$  points. A point  $p$  in  $S$  is **extremal** if there exists a line  $L$  that passes through  $p$  such that the remaining points of  $S$  all lie strictly to one side of  $L$ . The set  $S$  is **extremal** if every point of  $S$  is extremal.
    - (a) Outline an  $O(n \log n)$  algorithm to check whether  $S$  is extremal.
    - (b) Suppose we are given the points in  $S$  sorted by their  $x$ -coordinates. Can you solve the same problem in  $O(n)$  time?
  3. Exercise 33.4-1, p.961. What is wrong with Professor Smother’s scheme?
  4. Augmented Dijkstra Algorithm. Suppose that in the single-source shortest path problem, we not only want to determine the cost of the shortest path from source  $s$  to any node  $v$ , we also want to reconstruct one path with that cost.
    - (a) Indicate how to our hand-simulation of Dijkstra’s algorithm (which maintains a single array  $c[1..n]$  as in hw6) so that we maintain the shortest path information as well. HINT: read page 584 which describes using another array  $\pi[1..n]$ .
    - (b) Show this for the example in hw6.
  5. Problem 34.3-3, page 994.  $L \leq_P \bar{L}$  iff  $\bar{L} \leq_P L$ . Note that  $\bar{L}$  is the complement of the language  $L$ .
  6. Problem 34.2-3, p.983. Listing vertices of a Hamiltonian cycle.
  7. Problem 34.2-6, p.983. Hamiltonian Path is in  $NP$ .

---

## ANSWERS

---

1. Problem 33.1-4, page 939.

In Problem 33.1-3, we saw that we can perform a sorting of the polar angles of the input sequence  $S$  of  $n$  points about an arbitrary point  $O$  in  $O(n \log n)$  time. Call  $O$  the origin of this polar sorting. Suppose  $A, B, C$  are collinear, with  $B$  between  $A$  and  $C$ . If you perform a sorting of the polar angles using  $A$  as origin, it is easy to detect this collinearity. Hence, if we perform  $n$  polar sortings, using each of the points of  $S$  as origin, we can detect if there are any collinearity of three points.
2. Verifying Extremal Point Set. (a) Just use any convex hull algorithm that runs in  $O(n \log n)$  time. Then  $S$  is extremal iff the convex hull of  $S$  has  $n$  points.

(b) If  $S = (p_1, \dots, p_n)$  is already sorted by the  $x$ -coordinates, then the convex hull of  $S$  can be broken up into two convex chains, both running from  $p_1$  to  $p_n$ . One chain is convex upwards, and the other is convex downwards. Call them the upper and lower chains, respectively. For each  $i = 2, \dots, n - 1$ , we call  $p_i$  a candidate for the upper chain if  $p_i$  lies above the line through  $p_1, p_n$ . Otherwise it is a candidate for the lower chain. We now just have to check that the subsequence of candidates for the upper chain is really convex upwards. Similarly for the lower chain. This checking takes  $O(n)$  time.

3. Exercise 33.4-1, p.961. Actually, there can be 7 points in the rectangle. How?
4. Augmented Dijkstra Algorithm. The array  $\pi[1..n]$  records for each vertex  $i$  where the “current” shortest path from source to  $i$  can go through just before reaching  $i$ . Thus  $(\pi[i], i)$  is the last edge in a shortest path to  $i$ . The edge before that is  $(\pi[\pi[i]], \pi[i])$ , and so on. Thus we can reconstruct the entire path to  $i$ .
5. Problem 34.3-3, page 994.  $L \leq_P \bar{L}$  iff  $\bar{L} \leq_P L$ . Suppose  $L \leq_P \bar{L}$ . By definition, it means there exists some polynomial-time computable function  $t$  such that  $x \in L$  iff  $t(x) \notin \bar{L}$ . But this means  $x \notin L$  iff  $t(x) \in \bar{L}$ . Thus,  $x \in \bar{L}$  iff  $t(x) \in L$ . This proves that  $\bar{L} \leq_P L$  (via the same function  $t$ ).
6. Problem 34.2-3, p.983. Listing vertices of a Hamiltonian cycle.

We want a polynomial time algorithm  $A$ , which given a graph  $G$ , with either say that  $G$  has no Hamiltonian cycle, or else list the vertices in a Hamiltonian cycle of  $G$ . Let  $B$  be the polynomial time algorithm that checks if an input graph belongs to *HAM – CYCLE*.

We may assume that  $G$  does have a Hamiltonian cycle (why?). Pick an arbitrary starting node  $v_1$ . Inductively, assume that we have found a sequence  $(v_1, v_2, \dots, v_i)$  of vertices that can be completed into a Hamiltonian cycle. Pick a new vertex  $u$ . We want to know whether we can make  $u$  into  $v_{i+1}$ . To do this, we construct a graph  $G'$  in which all the vertices  $v_1, \dots, v_i$  and  $u$  have been replaced by a new vertex  $u^*$  (all other vertices remain intact). All edges involving the old vertices are also intact. We introduce a new edge  $(u^*, v)$  whenever there is an old edge  $(u, v)$ , for all  $v$  that remain. We also introduce a new edge  $(v, u^*)$  whenever there is an old edge  $(v, v_1)$  for all  $v$  that remain. It is easy to see that:

There is a Hamiltonian cycle in  $G'$  that extends  $(v_1, \dots, v_i, u^*)$  iff there is a Hamiltonian cycle in  $G$  that extends  $(v_1, \dots, v_i, u)$ .

If  $B$  rejects  $G'$ , then we must try another choice of  $u$ . Eventually we succeed. Thus we can find a candidate for  $u_{i+1}$  with at most  $n$  calls to  $B$ . Thus in  $n^2$  calls to  $B$ , we can find a Hamiltonian cycle for  $G$ . Clearly this is in polynomial time.

7. Problem 34.2-6, p.983. Hamiltonian Path is in *NP*. This is very simple idea: we need to show a verifier  $V$  for *HAM – PATH*. Input for the verifier is a pair  $(G, P)$  and the verifier will check if  $P$  is a Hamiltonian path for  $G$ . This is clearly in polynomial time. To prove that this is a verifier, we need to show: if  $G \in \text{HAM – PATH}$  then there is some  $P$  such that  $V(G, P) = 1$ . Conversely, if  $G \notin \text{HAM – PATH}$  then  $V(G, P) = 0$  for all  $P$ . But this is obvious.