Solution to Homework 4

Yangwei Liu

November 28, 2012

1 Problem 1

Find an arbitrary point on the outer circle, and find the tangent to the inner circle. Suppose the tangent line intersect with the outer circle at point B, then continue to find tangent to inner circle from B. Repeat this until we form a polygon.

This is because the tangent line will cover the largest angle. And we have 2π to cover.

2 Problem 2

A minimum spanning tree is a connected graph, so for any cut of the original graph (X, Y) where $X \cup Y = V$,we have one edge connecting from X to Y in the MST. And as it is a minimum spanning tree, we can only choose the edge with minimum weight from the edges connecting from X to Y. And as the weights are distinct, we have an unique choice for each cut. So enumerate all cuts for the graph, each time we get an unique edge for the MST. So the MST is unique.

3 Problem 3

As T is a spanning tree, so deleting e will disconnect T into two connected components, V_1 and V_2. Now in $T^{'}$, V_1 and V_2 must be connected, so there is an edge $e^{'}$ connecting them. Now let's prove $e \neq e^{'}$. First we prove in a spanning tree of G, there is one and only one edge connecting V_1 and V_2. Suppose we have two, which are (x_1, y_1) and (x_2, y_2), and $x_1, x_2 \in V_1$, $y_1, y_2 \in V_2$. Because x_1 and x_2 are connected, and so do y_1 and y_2, this will form a circle. So because e is not in $T^{'}$, we have $e \neq e^{'}$. The rest easily follows.

4 Problem 4

Order the n files in increasing order.

This is because any other order will increase average time at some point. Suppose instead of $\{1,2,3,4,5\}$, you have $\{1,2,4,3,5\}$, then beginning from the third element, you get a larger average time, until the end.

5 problem 5

Note that we want to design an $O(n)$ algorithm, so sorting is not allowed. The algorithm is as follows:

Denote the value of items as v_i, denote weight as w_i. Denote knapsack capacity as W.

First compute $\frac{v_i}{w_i}$ for all i. Use SELECT algorithm to find the median. Denote as m. Then divide items into 3 sets: $G = \{i \mid \frac{v_i}{w_i} > m$, $E = \{i \mid \frac{v_i}{w_i} = m$ and $L = \{i \mid \frac{v_i}{w_i} < m$. Compute $W_G = \sum_{i \in G} w_i$ and $W_E = \sum_{i \in E} w_i$.

1. if $W_G > W$, we do not take any item from G. We continue to divide G.
2. if $W_G \leq W$, we take all items from G, and take as many items as possible from E.
3. if $W_G + W_E \geq W$, we are finished;
4. otherwise, continue to find solution on L, with W updated to $W - W_G - W_E$

Notice that the recursion formula for the algorithm is $T(n) = T(\frac{n}{2}) + O(n)$, because each time we can handle at least half of the items. So by Master Theorem, $T(n) = O(n)$.

6 Problem 6

Let $d(u)$ denote the degree of vertex u. The main difficulty is to locate one of the interesting vertices(the body, the tail or the sting); after that we can locate all other vertices in $3n$ probes.For example, if we have found a vertex v with $d(v) = n - 2$, then that vertex must be the body if the graph is a scorpion. By scanning the v'th row of the matrix we can check that $d(v) = n - 2$ and determine its unique non-neighbor u, which must be the sting if the graph is a scorpion. Then by scanning the u'th row, we can verify that $d(u) = 1$ and find its unique neighbor w, which must be the tail; and with n more probes we can verify that $d(w) = 2$.

We start with an arbitrary vertex v and scan the v'th row. If $d(v) = 0$ or $n - 1$ the graph is not a scorpion. If $d(v) = 1, 2$ or $n - 2$, then either v is interesting itself or one of its 1 or 2 neighbors is, and we can determine all the interesting vertices as above and check whether the graph is a scorpion with at most 4n additional probes.

Otherwise,$3 \leq d(v) \leq n - 3$,and v is boring. Let B be the set of neighbors of v and let $S = V - (B \cup \{v\})$. The body must be in B and the sting and tail must be in S. Choose arbitrary $x \in B$ and $y \in S$ and repeat the following: if

x and y are connected, the delete y from S(y cannot be the sting) and choose a new $y \in S$. If x and y are not connected, then delete x from B(x is not the body unless y is the sting) and choose a new $x \in B$. If the graph is indeed a scorpion, then when this process ends, B will be empty and y will be the sting. To see this, observe that B cannot be emptied without encountering the sting, because the body cannot be deleted from B by any vertex in S except the sting; and once the sting is encountered, all remaining elements in B will be deleted. Whether or not the graph is a scorpion, the loop terminates after at most n probes of the adjacency matrix, since after each loop some vertex is discarded.