(1) Data Structures and Algorithms (for Part 1b only) 2000

p6q1
MR

Describe an $O(n\log(n))$ algorithm based on a variation of merge sort to find the closest pair of a given set of points lying in a plane. You may assume the set of points is given as a linked list of x-y coordinates. \marks{8}

Carefully prove that you algorithm can never take longer than $O(n\log(n))$.\marks{6}

Modified, with explanation, your algorithm to find the pair of points with minimum Manhatten distance. The manhatten distance between points (x_1,y_1) and (x_2,y_2) is $|x_1-X_2|+|y_1-y_2|. \marks{6}

Answer:

Initialise the p1, p2 and mindist to be unknown, ubknown, maxint Use merge sort to sort the points in on the x coordinate. Use a modified version of merge sort to this list on the y coordinate, while at the same time updating p1, p2 and mindist as closer pairs of points are found. This only requires a change in the merge operation which is bookwork. Only at most $4n$ distance comparisons a needed when two lists are merged to produce a list of length n. Prove od $O(n\log(n))$ is book work. Changing to manhatten distances only changes the distance function, everything else works unchanged.

(2) Data Structures and Algorithms

Describe an efficient algorithm based on Quicksort that will find the element of a set that would be at position k if the elements were sorted. \marks{6}

Describe another algorithm that will find the same element, but with a guarateed worst case time of $O(n)$. \marks{8}

Give a rough estimate of the number of comparison each of your methods would perform when $k=50$ and a set of 100 random integers. \marks{6}

Answer:
Bookwork
Bookwork using the quintuple algorithm
last part not difficult

(3) Data Structures and Algorithms

Describe in detail both Prim's and Kruskal's algorithms for finding minimum cost spanning tree of an undirected graph with edges labelled with positive costs. \marks{7 each}

Compare the relative merits of the two algorithms. \marks{6}

Answer:
Bookwork

(4) Data Structures and Algorithms

Explain what is meant by the terms {\em directed graph}, {\em undirected graph} and {\em bipartite graph}. \marks{3}

Given a bipartite graph what is meant by a {\em matching}, and what is an {\em augmenting} path with respect to a matching. \marks{4}

Prove that if no augmenting path exists for a given matching then that matching is maximal. \marks{6}

Outline an algorithm based on this property to find a maximal matching, and estimate its cost in terms of the number of vertices n and edges e of the given bipartite graph. \marks{7}

Answer: