高等演算法 第 4 次作業

1. For two points p and q in the plane, we say that p ***dominates*** q if both x and y coordinates of p are no less than that of q respectively. Given a set S of n points, the ***rank*** of a point p in S is the number of points in S dominated by p. We want to find the rank of every point in S. A naïve algorithm needs $O(n^2)$ time. Design a faster algorithm for this problem.

2. Assume there are n supposedly identical VLSI chips that are capable of testing each other. There is a test jig that can accommodate two chips at a time. When the jig is loaded, each chip tests the other and reports whether it is good or bad. A good chip always reports accurately whether the other chip is good or bad, but the answer of a bad chip cannot be trusted. Show that the good chips can be identified with $O(n)$ pairwise tests, assuming that more than $n/2$ of the chips are good.

3. Let T and P be two sequences such that $|P| \le |T|$. Design a linear time algorithm to determine whether P is a subsequence of T.

4. A sequence $Z = z_1, z_2, \ldots, z_k$ is a subsequence of $X = x_1, x_2, \ldots, x_n$ if there exists a strictly increasing sequence $\langle i_1, i_2, \ldots, i_k \rangle$ of indices of X such that for all $j = 1, 2, \ldots, k$, we have $x_{i_j} = z_j$. For example, Z = bcdb is a subsequence of X = abcbdab with corresponding index sequence $\langle 2, 3, 5, 7 \rangle$. Design an algorithm that counts the number of occurrences of Z in X as a subsequence such that each has a distinct index sequence.

5. Given a sequence S of n nonnegative numbers $x_1, x_2, \ldots, x_n$, and an integer k, partition S into k or fewer consecutive subsequences such that the largest sum of these subsequences is minimized over all possible partitions.

6. You have inherited the publishing rights to n songs by the Raucous Rockers. You want to release a boxed set of d compact disks, each of which can hold at most m minutes of music. To satisfy the fans, you must put the songs in chronological order, but you can omit songs (regardless of when they were recorded) if necessary. Of course, no song can be split across a disk. Given a list of the song lengths in chronological order, your task is to figure out the maximum number of songs that can be recorded on the set of disks subject to these criteria.

7. Solve “Bitonic Euclidean traveling-salesman problem” in chapter 15 of the book [CLRS].

8. Consider the activity-selection problem appears in p.41 of unit-5. Now, assume that each given activity is associated with a positive weight. Design an algorithm to find an independent subset of activities (activities that can be allowed to use the same resource) whose sum of weights is maximized.

9. Given a list of n positive integers $d_1, d_2, \ldots, d_n$, we want to efficiently determine whether there exists a simple undirected graph whose nodes have degrees precisely $d_1, d_2, \ldots, d_n$. Design an algorithm to solve this problem.

10. Given a weighted graph, find a spanning tree such that the maximal edge weight of the tree is minimized over all spanning trees.