1 Given an array A of n distinct positive integers and an integer S, find the number of (unordered) pairs of indices {i, j} such that i is not equal to j and such that A[i] + A[j] > S, with an algorithm which runs in time O(n log n). Justify that your algorithm indeed runs in time
O(n log n). (20 pts.)

Fill in your answer here Help

2 A sub-graph H of a graph G consists of a subset of vertices of G together with all edges of G whose both ends are in H. Assume now that you are given an undirected graph G with N vertices by an adjacency matrix (i.e., a matrix M such that M(i, j)=1 if there is an edge with end points i and j and M(i, j)=0 otherwise). Design an algorithm which finds a largest size sub-graph H of G such that every vertex of H is adjacent (i.e., connected by an edge) with at least 3 other vertices of H and is also NOT adjacent with at least 3 other vertices of H; if there is no such sub-graph the algorithm should output a message “No such H can be found”. Estimate the run time of your algorithm as a function of the number of vertices N.

Fill in your answer here Help

Format

Words: 0

Maximum marks: 20

3 You are given a sequence A=(a(1), a(2), ..., a(n)) of length n, where all a(i) are positive integers. Consider all subsequences B of A which satisfy the condition that no two consecutive elements of A, say a(i) and a(i+1), both belong to B. Let sum(B) be the sum of all elements of such a subsequence B. Design a dynamic programming algorithm which finds the largest possible value of sum(B) when B ranges over all subsequences meeting the above condition. In particular, you have to:
(a) State clearly the sub-problems that need to be solved
(b) write explicitly the recursion equations for solving these sub-problems;
(c) explain how the final answer is obtained;
(d) estimate the run time of your algorithm.
Note that you are only asked to find the maximal possible value of such sums of elements of B satisfying the above condition; you do not have to find B itself, just the sum of its elements.

Fill in your answer here Help

Format

Words: 0

Maximum marks: 20

4 You are given a rooted tree T with n vertices. Each vertex is coloured either blue or green.
(a) Your first task is to design a dynamic programming algorithm which for every sub-tree T(i), rooted at a vertex i of T, finds the LENGTH of the longest possible path through the sub-tree T(i) which starts at the root i and such that all vertices on this path are of the same colour as the root i. Your algorithm MUST run in time O(n). (8 points)
(b) Your second task is to design a dynamic programming algorithm which finds the LENGTH of the longest possible path through the tree (not necessarily starting at the root) such that all vertices on this path are of the same colour, either all blue or all green. (12 points)
In particular, for both problems separately, you have to:
(a) state clearly what the sub-problems are that need to be solved;
(b) describe precisely the recursion for solving these sub-problems;
(c) explain how the final answer is obtained;
(d) estimate the run time of your algorithm.
Note that you are only asked to find the length of such a path and not the path itself.
Also, you can use the solution of problem (a) to help you obtain a solution of problem (b).

Fill in your answer here Help

Format

Words: 0

5 Use a max flow algorithm to solve the following problem. Elbonia is grappling with a COVID outbreak. It has N affected cities 1,…,N. City i has p(i) many patients who require an intensive care bed. Hospitals in a city i have b(i) many available intensive care beds in total. Transportation of patients living in a city i to a hospital in a city j where they will be treated is possible only if the city j is within K kilometers (by road) from city i. You are given the above numbers and a map of Elbonia with road distances and have to design an algorithm which assigns each patient from every city i to a sufficiently close city j which has a hospital where that patient can be treated. Your algorithm should make sure that no city i has to accommodate more patients than its total intensive care unit capacity b(i). It should also output a message "impossible" if there is no such assignment. In particular:
(a) describe what the vertices of the flow graph are;
(b) describe what the edges of the flow graph are;
(c) describe how the capacities of all edges are assigned;
(d) after applying a max flow algorithm describe how the patients are assigned to cities where they will be treated;
(e) describe how it is determined that the problem has no solution meeting the constraints given.
Note: p(i) can be larger or smaller or equal to b(i); also a patient from a city c(i) does NOT have to be treated in a hospital in the same city.

Fill in your answer here Help

Format