MinorII (COL 351)

Read the instructions carefully:

- The first two questions can be done by dynamic programming. You need to clearly state the meaning of the entries in the table, show the recurrence being used (no need to prove it, but 1 or 2 lines of justification need to be given) along with the base case, and finally show some sort of pseudo-code which illustrates the orders in which the entries are computed. Finally, analyze the running time of your algorithm. In both the questions, the input is a sequence of numbers. You can assume that these sequences are given in an array. Further, the output is just a number in both the questions.
- The last question requires a proof. The proof should be written concisely without referring to any examples.

1. You are given a sequence of integers (which could be positive or negative). Between every two consecutive integers, you need to place either the plus '+' operator or the product '*' operator such that the value of the resulting expression is maximized. Note that there are no brackets here – once you place the '+' and '*' signs, the expression is evaluated using the usual precedence laws – multiplication precedes addition. For example, suppose the numbers are $6, -1, 2, -3$. If you place operators like, $6, +, -1, *, 2, +, -3$, then the result is 1. If you place them like $6, *, -1, *, 2, +, -3$, the result is -15. If they are placed as $6, +, -1, +, 2, *, -3$, the result is -1. Give an efficient algorithm which outputs the maximum value of such an expression. **(7 marks)**

2. You are given a sequence of n integers, call them $a_1, a_2, \ldots, a_n$. Further, for each integer a_i, you are given a positive weight w_i. We would like to arrange them in a **binary search tree** such that the quantity $\sum_{i=1}^{n} w_i \cdot h_i^2$ is minimized, where h_i is the depth of the integer a_i (the depth of root is 1, its children have depth 2, and so on). Observe that the objective function has **squares** of the depths of these numbers. Give an efficient algorithm which outputs the minimum value of $\sum_{i=1}^{n} w_i \cdot h_i^2$. **(7 marks)**

3. Consider a bipartite graph where the vertex sets on the two sides are denoted by A and B respectively. Let A_1 be a subset of A and B_1 be a subset of B. Suppose there are two matchings M and M' in the graph such that M matches all the vertices in A_1 and M' matches all the vertices in B_1 (recall that a matching is said to match a vertex v if there is an edge in the matching which has v as an end-point). Prove that there is a matching in the graph which matches all the vertices in $A_1 \cup B_1$. **(6 marks)**

M' $W_i \ldots W_{t-1}$ W_t $W_{t+1} \ldots W_n$

$i \to n$ $a * b +$ $a +$ B_1 W_t

$a_i * a_{i+1} \ldots$ a_n

$a +$ ____

$a *$

$W_t \cdot 1^2 +$ $\sum_{j=i}^{t-1} w_j (1+h)^2$

$1 + h^2 + 2h$

$(\ ) + (\ )$ $a + (\ )$ $a * b * c * d$

$B \to$ it already Matched $W_t \cdot 1^2 + \sum_{j=i}^{t-1} w_j + \sum_{j=i}^{t-1} w_j \cdot 2h + \sum$

with $A_1 \to$ done

else in out $\to$ done $\sum_{k=i}^{j} w_k +$ $dp[i - t-1]$

$+ dp[t+1 \to j]$

$a * b * c * d +$ $a +$ ____ Not matched $a \in A$, $b \in B_1$ $+ 2[dp[i \to t-1]$

i $i+1$ $i+2$ k $M' \to (a, b)$ $+ dp[t+1 \to j]$

a $a \in A_1$ Problem

$a * a \to$