Question #1: Planning a company party (Problem 15-6 from CLRS)

You are asked to plan the company party for a big company and the first step is to choose who will be invited. There are two factors to be considered: a) each person has a desirability a_i; if this is high you really want to invite the person to the party (think the funny guy); b) the company hierarchy is organized as a rooted the tree. The root is the CEO of the company and everyone else is the child of their direct boss in the tree. You should never invite someone and their direct boss to the party (that would make it hard for them to have fun.)

Your job is to devise an algorithm that takes n, the number of people in the company, $a_1, \ldots, a_n$ and the company hierarchy as input and outputs the set of people to invite. The way the company hierarchy is given is as follows. You are given sets $C_1, C_2, \ldots, C_n$ where C_i contains the direct subordinates of the ith employee. You are guaranteed that these make a tree (each person except the CEO is in precisely one of the C_i sets and there are no cycles.) To make your life easier we also know that the CEO is always employee number 1 and that each person's subordinates have a higher number than themselves.

You should never invite a person and their direct boss at the same time and among all possible such solutions you should output the one with the maximum total desirability. The total desirability of an invitation list is the sum of the desirability of the people invited.

Example: An example input is $n = 4$; $a_1 = -2$, $a_2 = 1$, $a_3 = 1$, $a_4 = 4$; $C_1 = \{2, 3\}$, $C_2 = \{4\}$, $C_3 = \emptyset$, $C_4 = \emptyset$. The company hierarchy is drawn below in Figure 1.

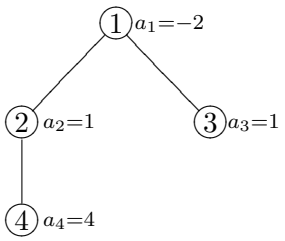

Figure 1: An example of a company hierarchy.

There are 8 valid invitation lists (ones that respsect the rule about not inviting a person and their direct boss): $\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 4\}, \{2, 3\}, \{3, 4\}$. The total desirabilities of these invitation lists are $0, -2, 1, 1, 4, 2, 2, 5$ respectively. So the optimal solution is to invite the employees 3 and 4 and no one else.

Solution: (Please demonstrate the solution on a tree.)

Define the following two quantities:

$$D[i] = \text{“The maximum possible total desirability of a valid invitation list that only invites } i \text{ and its (direct or indirect) subordinates.”},$$

$$Dexclude[i] = \text{“The maximum possible total desirability of a valid invitation list that only invites the (direct or indirect) subordinates of } i\text{.”}.$$

It is not hard to see that these two values satisfy the following recursive conditions.

$$Dexclude[i] = \sum_{j \in C_i} D[j],$$

$$D[i] = \max(\sum_{j \in C_i} D[j], a_i + \sum_{j \in C_i} Dexclude[j]).$$

It is also clear from definition that $D[1]$ is what we are trying to compute. So we can write two procedures that compute these values recursively. Instead observe that $D[i]$ and $Dexclude[i]$ depend only on $D[j]$ and $Dexclude[j]$ when j is a subordinate of i so we can compute these values in a buttom up fashion on the company hierarchy tree. Given that we know all the subordinates of i have bigger numbers that i we can do this in an even easier way: compute $D[n], Dexclude[n]$ first then $D[n-1], Dexclude[n-1]$ and all the way down to $D[1], Dexclude[1]$.

Implementation:

Input: Integer n, desirabilities $a_1, \ldots, a_n$ and subordinate lists $C_1, \ldots, C_n$.
Output: The maximum possible total desirability.

```
1  int D[1...n]
2  int Dexclude[1...n]
3  for i ← n downto 1 do
4      sum_of_children ← 0
5      sum_of_children_excluded ← 0
6      foreach j ∈ C_i do
7          sum_of_children_excluded ← sum_of_children_excluded + Dexclude[j]
8          sum_of_children ← sum_of_children + D[j]
9      end
10     Dexclude[i] ← sum_of_children
11     D[i] ← max(sum_of_children_excluded + a_i, sum_of_children)
12 end
13 return D[1]
```

Run time: The ith iteration of the for loop takes $2|C_i| + 4$ steps, so the total runtime is $4n + 2\sum_i |C_i|$. But each person except the CEO has one direct boss and the CEO has no direct boss so $\sum_i |C_i| = n - 1$. So the total runtime is $6n - 2 = \Theta(n)$.

Proof of optimality: The kth iteration of the for loop has $i = n - k + 1$ we prove by induction on k that $D[i], \ldots, D[n]$ and $Dexclude[i], \ldots, Dexclude[n]$ are correct after this iteration.

If you had extra time: Discuss how to compute the optimal invitation list (not just the maximum total desirability.)