

problema *Monete a posto (monete)*

```
#include <iostream>
using namespace std;
#include <fstream>

int A [100001]; //--- monete di #i da 1 a N
int B [100001]; //--- padre di #i da 2 a N

int N, dif=0, OP=0;

int main(){
    ifstream in ("input.txt");
    ofstream out ("output.txt");
    // lettura INPUT
    in >> N;

    for (int i=1; i<=N; i++)
        in >> A[i];
    B[1] = 0;
    for (int i=2; i<=N; i++)
        in >> B[i];

    for (int i=N; i>1; i--)
        if (A[i] != 1) {
            dif = A[i] - 1;
            A[B[i]] = A[B[i]] + dif;
            A[i] = 1;
            if (dif > 0)
                OP += dif;
            else
                if (dif < 0)
                    OP -= dif;
        } //--- nodi con 1 moneta non elaborate- esempio #8

    out << OP;
    return 0;
}
```

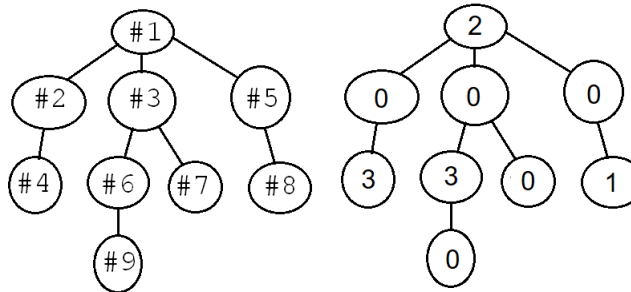
valore di OP all'inizio del ciclo for →

Caso di test 1

9
 2 0 0 3 0 3 0 1 0 ← da A [1] a A [9]
 1 1 2 1 3 3 5 6 ← da B [2] a B [9]

Grafo indicato da B []

valori in A []



valori in A[] e in OP al decrescere di i nel ciclo for →

A partire dall'ultimo nodo e risalendo fino alla radice, si elaborano i soli nodi con monete maggiori di 1 o inferiori ad 1, ponendole ad 1 ed incrementando il numero operazioni eseguite del valore tolto al nodo i (se il numero di monete è maggiore di 1) o del numero di monete spostate su i (1 se il nodo conteneva 0 monete, più di 1 se il nodo aveva accumulato un debito negativo per riempire i nodi sottostanti). Viene aggiornato anche il nodo padre aggiungendogli le monete prestate da i (dif è positivo) o prestate ad i (dif è negativo).

OP=0	-----	i=N #9 da 0 a 1	dif= -1
	2		
	0 0 0		
	3 2 0 1		A[B[9]]=A[6]=3-1=2
	1		A[9] = 1
	---per i = 8 : A[8]=1 la condizione dell'if è FALSA---		
OP=1	-----	i=N-2 #7 da 0 a 1	dif= -1
	2		
	0 -1 0		A[B[7]]=A[3]=0-1=-1
	3 2 1 1		A[7] = 1
	1		
OP=2	-----	#6 da 2 a 1	dif= +1
	2		
	0 0 0		A[B[6]]=A[3]=-1+1=0
	3 1 1 1		
	1		
OP=3	-----	#5 da 0 a 1	dif= -1
	1		A[B[5]]=A[1]=2-1=1
	0 0 1		
	3 1 1 1		
	1		
OP=4	-----	#4 da 3 a 1	dif= +2
	1		
	2 0 1		A[B[4]]=A[2]=0+2=2
	1 1 1 1		
	1		
OP=6	-----	#3 da 0 a 1	dif= -1
	0		A[B[3]]=A[1]=1-1=0
	2 1 1		
	1 1 1 1		
	1		
OP=7	-----	#2 da 2 a 1	dif= +1
	1		A[B[2]]=A[1]=0+1=1
	1 1 1		
	1 1 1 1		
	1		
OP=8	-----	valore di OP a fine programma	