Grattacieli commemorativi (grattacieli)

In occasion of the XIX edition of the Italian Olympics in Informatics, the council of Matera decided to build N skyscrapers, each of them assigned to a different builder. The height of each skyscraper is set by Monica, as the supreme authority in the Olympiads, with the goal of building the highest possible skyscrapers while keeping everyone happy.

According to the budget of the i-th builder, the i-th skyscraper can be built up to a maximum height of $H[i]$. Moreover, some builders are worried of looking bad when compared to their competitors, so they decided to set limits to how much the other skyscraper can "surpass" theirs.

The builders set M constraints. Each constraint j is given as three numbers A_j, B_j, C_j and specifies that the owner of the A_j-th skyscraper won't accept that the B_j-th skyscraper surpasses A_j in height by more than C_j. If we call $S[i]$ the assigned heights, the following must be true: $S[B_j] \leq S[A_j] + C_j$.

Help Monica pick the heights of each skyscraper such that the **sum of their heights** is maximized, provided that all the constraints stated above are satisfied.

Implementation

You should submit a single file, with either a `.c` or `.cpp` extension.

> ☞ Among the attachments in this task you will find a template `grattacieli.c` or `grattacieli.cpp` with a sample implementation.

You will have to implement the following function:

C	`long long costruisci(int N, int M, long long* H, int* A, int* B, int* C);`
C++	`long long costruisci(int N, int M, vector<long long>& H, vector<int>& A, vector<int>& B, vector<int>& C);`

- The integer N is the amount of skyscrapers that need to be built.
- The integer M is the number of contraints imposed by the builders.
- The array H, indexed from 0 to $N-1$, contains the maximum height of each skyscraper.
- The arrays A, B and C, indexed from 0 to $M-1$, contain the constraints imposed by builders.

The `costruisci` function should return the maximum total height that can be obtained while satisfying every constraint.

Sample grader

Among this task's attachments you will find a simplified version of the grader used during evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the functions that you should implement and writes back on `stdout` using the following format.

The input file is formed by $M+2$ lines, containing:

- Line 1: the integers N and M.
- Line 2: the N integers $H_0, \ldots, H_{N-1}$.
- Lines $3, \ldots, M+2$: the three integers A_j, B_j and C_j that represent the j-th constraint.

The output file is formed by a single line, containing the value returned by `costruisci`.

Constraints

- $1 \leq N, M \leq 100\,000$.
- $1 \leq H_i \leq 10^{12}$.
- $0 \leq C_j \leq 10^9$ for each j.
- $0 \leq A_j, B_j \leq N - 1$ for each j.
- There are no duplicate constraints (there are no distinct j, k such that $A_j = A_k$ and $B_j = B_k$).
- There are not constraints from a builder towards himself (such that $A_j = B_j$).

Scoring

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all testcases of which it is formed.

- **Subtask 1 [0 points]**: Sample testcases.
- **Subtask 2 [4 points]**: $N \leq 5$ and the maximum heights are all ≤ 5.
- **Subtask 3 [9 points]**: $M = N - 1$ and $B_j = A_j + 1$ for each i.
- **Subtask 4 [8 points]**: $B_j > A_j$.
- **Subtask 5 [21 points]**: $N \leq 300$.
- **Subtask 6 [16 points]**: C_j is 0 or 1, for each j.
- **Subtask 7 [24 points]**: $N \leq 2000, M \leq 10000$.
- **Subtask 8 [18 points]**: No limits.

Examples

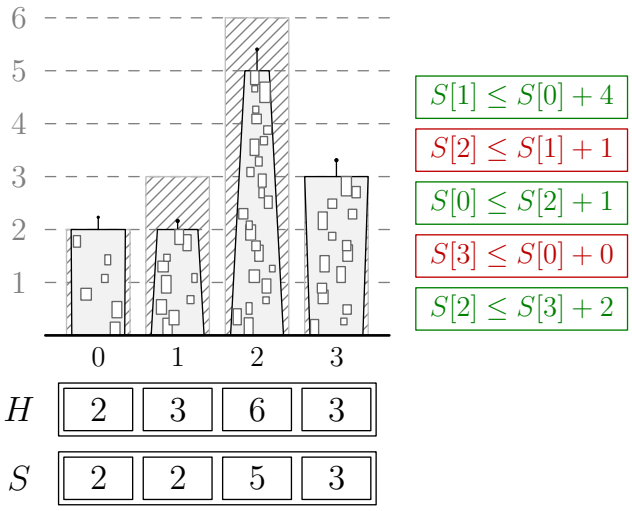
stdin	stdout
4 5 2 3 6 3 0 1 4 1 2 1 2 0 1 0 3 0 3 2 2	11
4 6 2 4 10 7 0 1 1 1 2 3 1 3 2 3 2 2 3 0 0 0 3 4	16

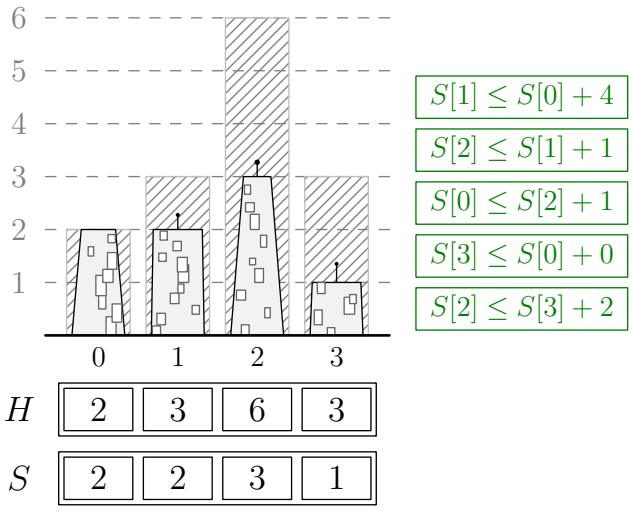
stdin	stdout
10 9 3 8 9 6 9 1 6 7 7 9 3 4 1 0 1 2 4 0 4 5 0 1 8 0 0 8 2 1 1 8 2 7 9 1 6 7 2	54

Explanation

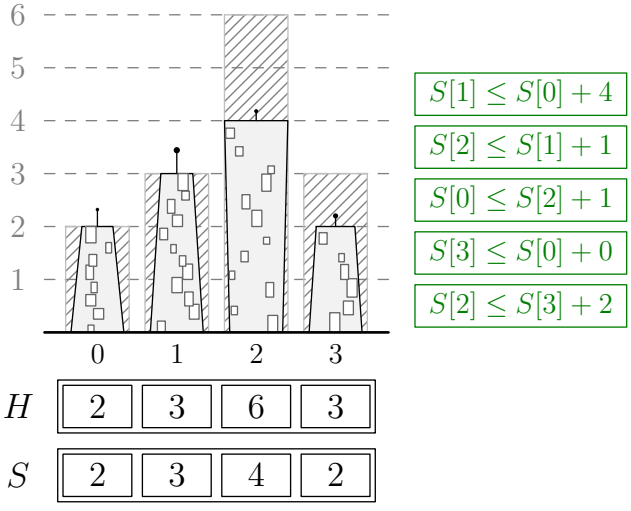
In the **first sample testcase** there are 4 skyscrapers and 5 contraints.

Here are two choices for the skyscrapers' heights. In the left one, the sum of heights is 12, but not all constraints are satisfied. In the right one, all constraints are satisfied but the sum of heights (8) is not the maximum.

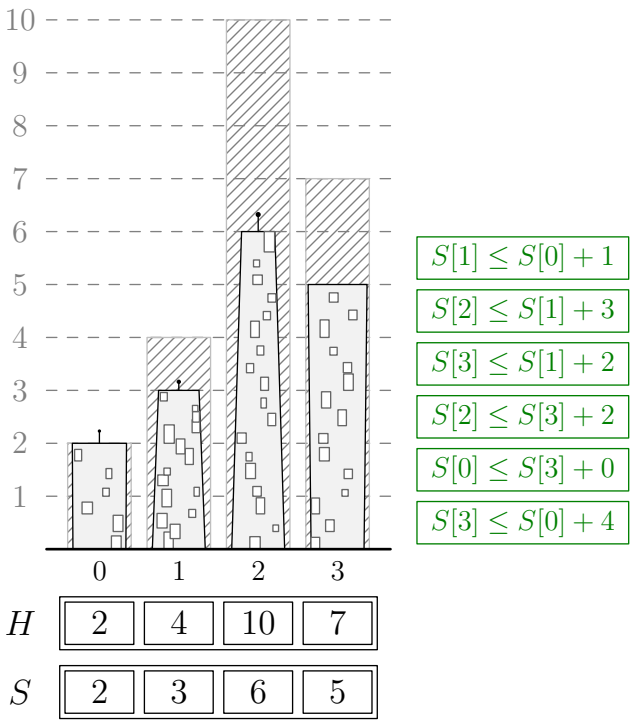




The maximum sum of heights possible is 11, and it can be obtained by choosing the following heights.



In the **second sample testcase** the maximum sum of heights is 16, and it can be obtained by choosing the following heights.



In the **third sample testcase** the maximum sum of heights possible is 54, and it can be obtained by choosing the following heights.