340 Lab - Highest Hanoi Tower

This is a variant of the Hanoi Tower game. You might have heard about the Tower of Hanoi that moves disks from one rod to another. This lab is a game for a different purpose.

Given a set of disks of different height and radius, you need to build the tower of Hanoi by stacking disks on top of each other. In order to put disk A on top of disk B, the radius and height of A must be strictly smaller than those of B.

Input:
A vector of disks. Each disk is a pair that contains two positive integer number corresponding to the radius and height of the disk respectively.

Output:
For each test case return:

1. the maximum height of the tower possible.
2. The index # of the disk in the input disk array that is on the top of the highest tower (index starts from 0).

<u>Your task</u>

Write a function that takes in a vector of disks, and returns height and index via the two reference values.

void buildHanoiTower(const vector< pair<int, int> > &v, int & height, int & index);

<u>File you need to submit</u>

Put your implementation of the function in a file called solution.cpp.

<u>Examples</u>

1. Two disks. One row per disk, with the first number being the radius and the second the height:

```
4 3
1 4
Max height of the Hanoi Tower: 4
Top disk index: 1
```

Explanation: The disk of radius 1 and height 4 is the only disk in the tower. So the tower height 4, the index of top disk is 1.

2. Three disks:

```
4 3
1 4
3 2
Max height of the Hanoi Tower: 5
Top disk index: 2
```

Explanation: The disk of radius 4 and height 3 is the disk at the bottom, and the disk of radius 3 and height 2 is stacked on top of it. So tower height is 3+2 = 5, the index of the top disk is 2, which is the third disk of (3 2).

3. Four disks:

```
2 6
1 4
4 5
2 2
Max height of the Hanoi Tower: 10
Top disk index: 1
```

Explanation: The disk of radius 2 and height 6 is the disk at the bottom, and the disk of radius 1 and height 4 is stacked on top of it. So the tower height is 6+4 = 10, top disk index is 1. Other combinations of disks would yield towers of lower height. For example, stack disk 3 on top of disk 2 is a valid tower, but the height would be 7.

<u>Additional notes</u>

Note 1: You can write your own main function to build a vector and test the output for various test cases. The testing code may look like below:

```
int height = 0;
int index = 0;
vector< pair<int, int> > v;
v.push_back(make_pair(4,3));    //you can add additional pairs here for more disks
for(auto disk : v)
        cout << disk.first << ' ' << disk.second << endl;

buildHanoiTower(v, height, index );
cout << "Max height of the Hanoi Tower: " << height << endl;
cout << "Top disk index: " << index << endl;
```

Note 2: Depending on the implementation and the actual disks, the max height is fixed, but the top disk index may have different results. The examples we give and the tests we do in the auto-grader have unique answers for top disk index.