

Project Euler #18: Maximum path sum I



This problem is a programming version of [Problem 18](#) from [projecteuler.net](#)

By starting at the top of the triangle below and moving to adjacent numbers on the row below, the maximum total from top to bottom is **23**. The path is denoted by numbers in bold.

```
  3
 7 4
2 4 6
8 5 9 3
```

That is, $3 + 7 + 4 + 9 = 23$.

Find the maximum total from top to bottom of the triangle given in input.

Input Format

First line contains T , the number of testcases. For each testcase:

First line contains N , the number of rows in the triangle.

For next N lines, i 'th line contains i numbers.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N \leq 15$
- $numbers \in [0, 100)$

Output Format

For each testcase, print the required answer in a newline.

Sample Input

```
1
4
3
7 4
2 4 6
8 5 9 3
```

Sample Output

```
23
```

Explanation

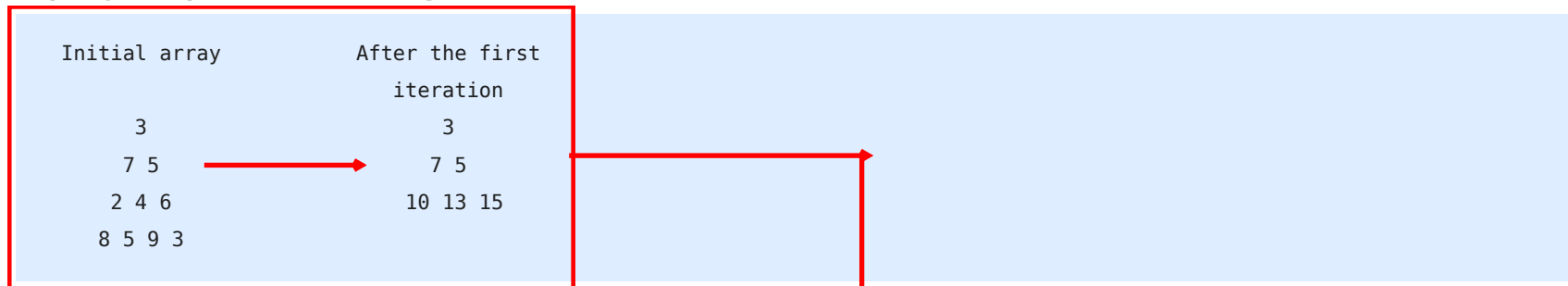
As shown in statement.

Solution

To solve this problem and [problem 67](#), which is much larger, start the search from the bottom to the top, adding the maximums along the way. This will “bubble” the maximum path total to the top of the triangle.

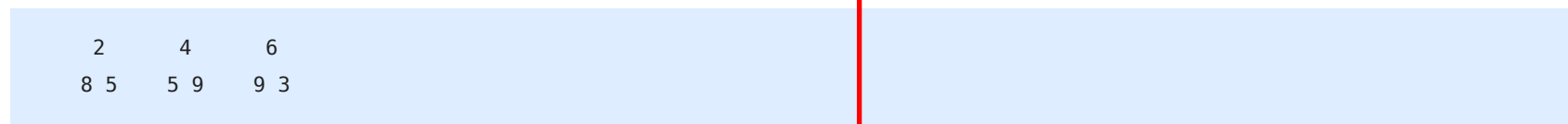
Let’s follow this technique, step-by-step, with the 4 row triangle example above to show how this works.

A step-by-step look at this algorithm



1. Starting at the bottom, take the first pair, 8 and 5, pick the maximum and replace the 2 in the previous row with their sum $8+2=10$.
2. Take the next pair, 5 and 9, pick the maximum and replace the 4 in the previous row with their sum $9+4=13$.
3. Finally, take the last pair, 9 and 3, pick the maximum and replace the 6 in the previous row with their sum $9+6=15$.

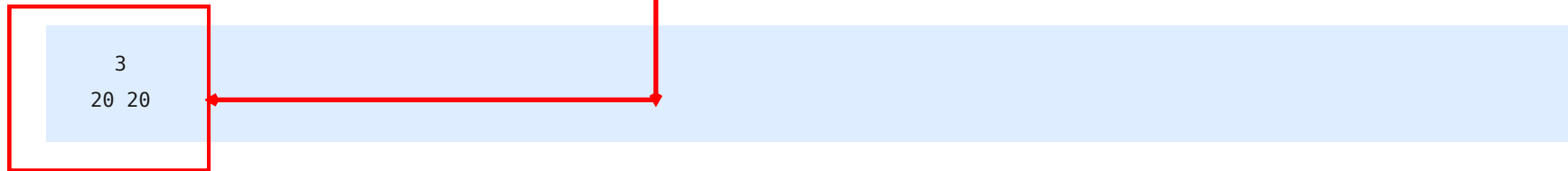
If you think about it, we simply solved the maximum path from the last row to the second to the last row by considering each sub-problem for the following three triangles.



Keep that in mind and let’s do it again with our new array.

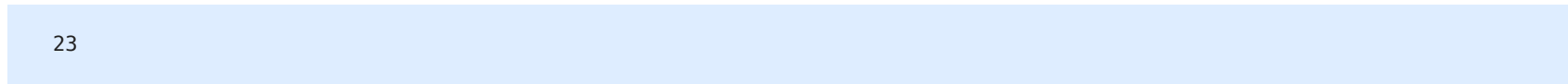
1. Take the larger of 10 and 13 and add it to 7 making $13+7=20$.
2. Take the larger of 13 and 15 and add it to 5 making $15+5=20$.

Now our array looks like:



At last we take the larger of 20 and 20 (yes, I know they're the same) and add it to 3 making $20+2=23$.

And our array looks like:



Which is the maximum total path in the triangle.

And if you follow this logic then you just witnessed dynamic programming in action—truly short and simple.

HackerRank version



Extended to solve all
test cases for Project
Euler Problem 18

HackerRank Project Euler 18 varies the number of rows in the triangle from $1 \leq N \leq 15$ and runs 10 test cases. No changes required except to read from std input instead of a file.

Python Source Code

```
1 table = [list(map(int, s.split())) for s in open('pe18.txt').readlines()]
2
3 for row in range(len(table)-1, 0, -1):
4     for col in range(0, row):
5         table[row-1][col] += max(table[row][col], table[row][col+1])
6
```