

CS1020E: DATA STRUCTURES AND ALGORITHMS I

Lab 5 – Tux

(Week 9, starting 10 October 2016)

Readme



Tux¹ the penguin is wobbling in a straight path/"line" along snow-covered terrain. Tux does not wobble to the side, but he **moves in one direction**, besides moving up and down due to the terrain. Help Tux perform two subtasks:

1. *Efficiently* find the most food while keeping his risk low 50%
Use only a **queue** for this subtask, and NO other container/array, not even to read input
2. *Efficiently* find number of pairs of relative peaks throughout his journey 50%
Use only a **stack** for this subtask, and NO other container/array, not even to read input

Can you perform each subtask by just making one pass through the input data?

Problem 1 - Angry Bird



50%

"A hungry bird is an angry bird". The path that Tux walks along is divided into $2 \leq N \leq 1,000,000$ units. Each unit i along the terrain contains a certain number of fish $0 \leq F_i \leq 1000$ happily swimming about below the snow.

Tux can choose to dive through the snow into the water below, and travel a number of units, catching all the fish along the way, i.e. catch all the fish within the interval $[F_{\text{dive}}, F_{\text{surface}}]$. Tux can only **dive once** in the entire journey. He may dive and surface in the same unit.

However, there is no free lunch. In certain units, there is the risk that Tux himself becomes lunch to other sea creatures. In all the other units, there is no such risk. Help Tux **efficiently find** and output a line containing the **most fish** that he can get, while keeping his risk to at most $1 \leq T \leq N$. In other words, the part of the journey in which Tux is underwater should not have more than T units that places Tux at risk.

The first line in the input contains T .

Each of the subsequent N lines contains two tokens separated by a space:

The risk that the unit presents to Tux, either 'R' for risky, or 'C' for clear

The number of fish in that unit, F_i

Reminder: Use only a **queue** for this subtask, and NO other container/array, not even to read input

Tip: Design your algorithm *completely* before coding, and see how using a queue helps

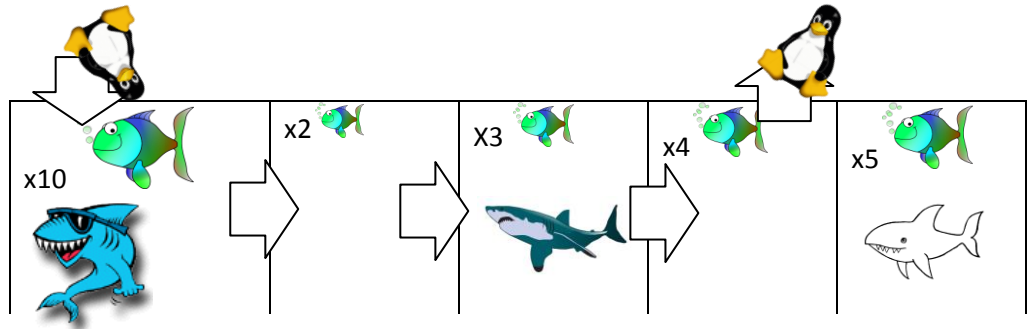
¹ Tux is the official Linux mascot. Picture credits: commons.wikimedia.org/wiki/File:Tux.png, clipartkid.com

Sample Input

```
2
R 10
C 2
R 3
C 4
R 5
```

Sample Output

```
19
```



Submission

Your source file should be named angrybird.cpp

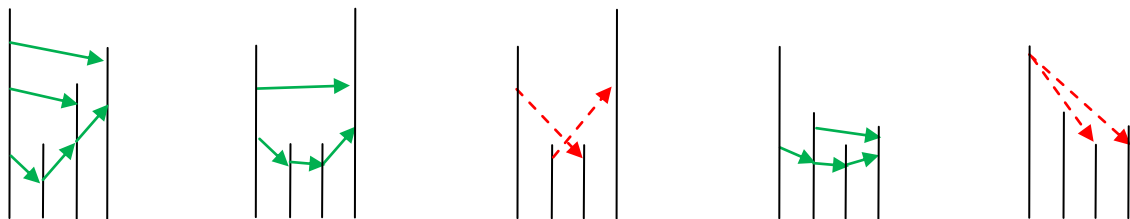
Problem 2 - Sightseer



50%

Tux has now eaten his fill. Walking up and down some mountains, Tux admires the beautiful scenery ahead of him. How many **pairs of relative peaks** are there?

Again, the terrain is divided into $2 \leq N \leq 1,000,000$ units, with each unit i having height H_i , a positive integer that fits within a 32-bit signed int. A pair of relative peaks is made up of two distinct units ($H_{\text{left}}, H_{\text{right}}$), where **left** < **right**, in which no unit in between is of the same height or is taller than either H_{left} or H_{right} .



In other words, either the left 'peak' is adjacent to the right 'peak', or every unit in between the two 'peak' is *strictly* lower than both the left and right 'peaks'. The input contains N lines, each representing H_i , in sequence.

Reminder: Use only a **stack** for this subtask, and **NO** other container/array, not even to read input

Sample Input

```
9
2
5
7
```

Sample Output

```
5
```

Submission

Your source file should be named sightseer.cpp

- End of Lab 5 -