CALSUCO '22

July 30th, 2022
120 Minutes
4 Problems
Problem Packet by CALICO

Table of Contents

Problem 1: Ice Cream Bars!
5+4+4=13 Points

Problem ID: `bars`
Rank: `1+2+3`

Introduction

Summer's finally here, and it's the perfect time to eat ice cream! For the past few months, you've been saving a stash of delicious ice cream bars for this very moment! You decide to come up with a plan to savor these tasty treats: on day 1, you eat 1 bar; on day 2, you eat 2 bars; on day 3, you eat 3 bars, and so on.

Oh, Haagen Dazs, if you wanna give us money, we're looking for sponsors ;)

Problem Statement

Given $\mathbf{N}$ bars of ice cream, find the number of days you can eat following your plan until you run out and won't be able to eat the full amount planned for the next day.

On day 1, you eat 1 bar. On day 2, you eat 2 bars. On day 3, you eat 3 bars, and so on. In other words, on day d, you eat d bars.

Input Format

The first line of input contains a positive integer $\mathbf{T}$ denoting the number of test cases that follow. Each test case is described in a single line containing a single integer $\mathbf{N}$ denoting the number of ice cream bars you have.

Output Format

For each test case, output a single line containing the number of days you can eat for before running out of ice cream bars and not being able to fully eat the next day.

Constraints

$1 \le \mathbf{T} \le 100$

Main Test Set

$0 \le \mathbf{N} \le 10^5$

Bonus Test Set 1

$0 \le \mathbf{N} \le 10^{15}$

Careful! If you are a Java or C/C++ programmer, be aware that the int variable type may be too small to contain $\mathbf{N}$! Java programmers can use variable types long or float instead, and likewise long long or float for C/C++.

Bonus Test Set 2

$0 \le \mathbf{N} \le 10^{10000}$

Careful! Values of $\mathbf{N}$ in this test set are extremely large! They exceed the maximum values of 64 bit integers and floats. This one can be quite tricky to get right, so we recommend trying other problems first if you're stuck.

Sample Test Cases

Sample Input

```
9
0
1
2
3
6
11
69
1337
12345
```

Sample Output

```
0
1
1
2
3
4
11
51
156
```

Sample Explanations

For test case 1, For test case 1, you have no ice cream bars. You can't eat any at all. Thus, you can only eat for 0 days.

For test case 2, you have 1 ice cream bar. You eat your only bar on day 1, and then won't have enough for day 2. Thus, you can only eat for 1 days.

For test case 3, you have 2 ice cream bars. You eat your first bar on day 1, and then won't have enough for day 2 because you need 2 but only have 1 more. Thus, you can only eat for 1 day.

For test case 4, you have 3 ice cream bars. You eat your first bar on day 1, and then your last 2 bars on day 2. You won't have enough for day 3. Thus, you can eat for 2 days.

For test case 5, you have 6 ice cream bars. You eat 1 bar on day 1, 2 bars on day 2, and 3 bars on day 3. You won't have enough for day 4. Thus, you can eat for 3 days.

For test case 6, you have 11 ice cream bars. You eat 1 bar on day 1, 2 bars on day 2, 3 bars on day 3, and 4 bars on day 4. Since you only have 1 left, you won't have enough for day 5. Thus, you can eat for 4 days.

Problem 2: Rock Paper Strategy!
0+10+12+14=36 Points

Problem ID: `rps`
Rank: `0+1+2+3`

Introduction

"If you know the enemy and know yourself, you need not fear the result of a hundred battles."
- *Sun Tzu, The Art of War*

Problem Statement

Play rock paper scissors against a bot running a specific program to win as many times as possible. A snippet of the source code of the program is given on the next page, but there are parts that are intentionally redacted.

Rock paper scissors is a two player game consisting of multiple rounds. In each round, both players make a move simultaneously: rock, paper, or scissors. Rock beats scissors, paper beats rock, and scissors beats paper. If your move beats your opponent's move, you win the round.

Although in actual rock paper scissors both players should move simultaneously, in this version, you'll move first, then the judge will move without considering your move for that round.

Bot Source Code

Other functions and variables may be present in the actual code but not in the snippet below.

```
function main():
    T = input_T()
    repeat T times:
        opponent.read_move();
        if opponent.last_move() == opponent.second_last_move():
            me.play(move_that_beats(opponent.last_move()))
        elif me.last_move() == me.second_last_move():
            me.play(move_that_beats(move_that_beats(me.last_move())))
        else:
            me.play(random_move())

X, A, C = get_3_random_nums_from_random.org()
M = 10 ** 9 + 7 # this is equivalent to 1000000007
q = []
function random_move():
    if q.length() == 0:
        X = (A * X + C) % M # % is the modulus operator
        temp = X
        repeat 19 times:
            q.add_to_end(temp % 3)
            temp = temp // 3 # integer division
    move = q.get_last_element()
    q.remove_last_element()
    if move == 0:
        return 'R'
    elif move == 1:
        return 'P'
    else:
        return 'S'
```

Input/Output Format

This is an interactive problem! Unlike regular problems, both your program and the judge will run simultaneously. Please refer to the contest guide for more information. Whenever you output, make sure to also flush your buffer as instructed by this post. If you have issues with buffering (or are getting time limit exceeded), please let us know via a clarification request! We'll be more than happy to help you out.

Begin by reading the first line of input containing the positive integer $\mathbf{T}$ denoting the number of test cases that follow. Each test case is a round of rock paper scissors. For each test case:

- First, output a single line containing a single character denoting your move for this round
  - This character should be one of `RPS`, representing rock, paper, or scissors respectively
- Then, read a single line of input containing a single character denoting the bot's move for this round
  - This character will be one of one of `RPS`, representing rock, paper, or scissors respectively

Constraints

$\mathbf{T} = 10^5$

Testing Test Set

This test set is impossible and is worth 0 points but serves for testing purposes. You can submit as many times as you want without accumulating any time penalty.
For this test set only, it is guaranteed that for the bot, `X`, `A`, and `C` have these values initially:
`X = 121292949, A = 653393711, C = 307210137`

Main Test Set

To pass, you must achieve a win rate of 30%.

Bonus Test Set 1

To pass, you must achieve a win rate of 60%.

Bonus Test Set 2

To pass, you must achieve a win rate of 90%.

Sample Interaction

Note that the line spacing here is to demonstrate the order in which interaction takes place only. Do not expect or output blank lines between each line of input or output.

Sample Input	Sample Output
7	
	P
P	
	S
R	
	S
S	
	S
R	
	P
R	
	S
S	
	S
P	

Sample Explanations

The judge begins by giving us $\mathbf{T}$ through the input. Note that for this sample, we have $\mathbf{T}$ = 7. The actual judge will always output $\mathbf{T}$ = 100000.

For round 1, we decide to play paper, so we output `P`. Then we read in the bot's move, which is also paper, `P`. Since we both played the same move, this round results in a draw.

For round 2, we decide to play scissors, so we output `S`. Then we read in the bot's move, which is rock, `R`. Since scissors beats rock, this round results in a loss.

For round 5, we decide to play paper, so we output `P`. Then we read in the bot's move, which is rock, `R`. Since paper beats rock, this round results in a win.

At the end of these 7 rounds, we have exactly 2 wins. This gives us a win rate of 2/7 ~= 28.5%, which means we would fail the test set just barely.

Problem 3:
Making the title
for this programming problem
was a lot of work
13+8=21 Points

Problem ID: `haiku`
Rank: `2+3`

Introduction

Anya wants to write a haiku for her best friend Damian, but language arts was never her strong suit! Unable to think of what words to use, she uses her telepathic abilities to read the minds of everyone around her to build a word bank. However, after coming up with these words, she remembers that math wasn't her strong suit either! Help Anya construct a haiku by using the words she collected!

Problem Statement

Given a list of $\mathbf{N}$ words represented by the number of syllables in each word $\mathbf{S_1}$, $\mathbf{S_2}$, ... , $\mathbf{S_N}$ as well as the words themselves $\mathbf{W_1}$, $\mathbf{W_2}$, ... , $\mathbf{W_N}$, construct a haiku.

A haiku is a three line poem with words containing 5 syllables in total on line one, 7 on line two, and 5 again on line three.

All words are unique and can only be used once.

If it's impossible to construct a haiku using the words provided, output the following instead:

```
IMPOSSIBLE
IMPOSSIBLE
IMPOSSIBLE
```

Input Format

The first line of input contains a positive integer $\mathbf{T}$ denoting the number of test cases that follow. For each test case:

- The first line contains an integer $\mathbf{N}$ denoting the number of words
- The next $\mathbf{N}$ lines contain 2 space-separated integers each $\mathbf{S_i}$ $\mathbf{W_i}$.
  - $\mathbf{S_i}$ denotes the number of syllables in the ith word
  - $\mathbf{W_i}$ denotes the letters of the ith word itself

Output Format

For each test case, output 3 lines:

- The first line contains the words of the first line of the haiku
- The second line contains the words of the second line of the haiku
- The third line contains the words of the third line of the haiku

Constraints

$1 \le \mathbf{T} \le 100$

$1 \le \mathbf{S_i} \le 9$

$1 \le |\mathbf{W_i}| \le 20$

$\mathbf{W_i}$ contains only letters from the lowercase alphabet: `abcdefghijklmnopqrstuvwxyz`

$\mathbf{W_i}$ may not be a real word in any language.

Main Test Set

$1 \le \mathbf{N} \le 9$

Bonus Test Set

$1 \le \mathbf{N} \le 1000$

Sample Test Cases

Sample Input

```
4
9
2 written
1 this
4 algorithm
4 definitely
1 was
1 by
1 an
2 haiku
1 cool
9
3 kawaii
3 shukudai
1 o
3 mogami
1 ni
2 roka
2 gawa
4 wasurete
2 moe
3
3 pentagon
5 dodecahedron
8 hecatonicosachoron
4
5 aaaaa
5 ccccc
7 bbbbbbb
9 ddddddddddddddddddddd
```

Sample Output

Note that this is only one of many possible correct outputs. Your program only needs to satisfy the syllable count requirements.

```
this cool haiku was
definitely written by
an algorithm
shukudai o
wasurete roka ni
mogami gawa
IMPOSSIBLE
IMPOSSIBLE
IMPOSSIBLE
aaaaa
bbbbbbb
ccccc
```

Sample Explanations

For test case 1, we are given 9 words. The haiku we construct has 1 + 1 + 2 + 1 = 5 syllables on line 1, 4 + 2 + 1 = 7 syllables on line 2, and 1 + 4 = 5 syllables on line 5. Since this follows the 5-7-5 syllable requirement, this is a valid haiku.

For test case 3, it's impossible to construct a haiku using the words provided, no matter how we arrange the words or place each word on different lines.

Problem 4: CALICO's Corporate Conundrum
15+9+6=30 Points

Problem ID: `managers`
Rank: `2+3+3`

Introduction

The year is 2069 and CALICO (which now stands for CALICO International Conglomerate) has expanded their operations! Compared to the two employees it had in 2022, CALICO now benevolently employs hundreds of thousands of low-wage contractors. With such a large company, disputes between the "people" working there often arise, and their managers need to step in to resolve them. Unfortunately, this often leads to managers being overworked!

Problem Statement

Find the largest number of unique disputes a single person is responsible for resolving within the company given the managers of all $\mathbf{N}$ of its employees. Each employee is identified using a number counting upwards from 0, with the last employee being assigned the number $\mathbf{N}$ - 1. The ith employee's manager $\mathbf{M}_i$ denotes the employee number of their direct superior. The CEO is assigned employee number 0 and is the only employee whose direct superior is themselves.

A dispute between two employees is resolved by the lowest-level manager that has authority over both of them, whether directly or indirectly. Any employee also has authority over themselves, meaning it's possible for a dispute to be resolved by someone involved in it. Two disputes are unique if they involve at least one different employee.

Input Format

The first line of the input contains a positive integer $\mathbf{T}$ denoting the number of test cases that follow. For each test case:

- The first line contains the single positive integer $\mathbf{N}$ denoting the number of employees present in the company.
- The second line contains the space-separated sequence of $\mathbf{N}$ non-negative integers $\mathbf{M_0}$, $\mathbf{M_1}$, …, $\mathbf{M_{N-1}}$ denoting the employee number of each individual's manager.
  - $\mathbf{M_0}$ will always be zero to represent the CEO being their own direct superior.

Output Format

For each test case, output a single line containing the largest number of unique disputes a single person is responsible for resolving within the company.

Constraints

$1 \le \mathbf{T} \le 100$

Main Test Set

$1 \le \mathbf{N} \le 50$
The sum of $\mathbf{N}$ across all test cases does not exceed 250.
An employee directly manages *at most* two other employees.

Bonus Test Set 1

$1 \le \mathbf{N} \le 10^5$
The sum of $\mathbf{N}$ across all test cases does not exceed 10^5.
An employee directly manages *at most* two other employees.
An employee has at most 10^3 direct or indirect managers.

Bonus Test Set 2

$1 \le \mathbf{N} \le 10^5$
The sum of $\mathbf{N}$ across all test cases does not exceed 10^5.
An employee can directly manage *any* number of employees.
An employee has at most 10^3 direct or indirect managers.

Sample Test Cases

Main Sample Input

```
4
5
0 0 0 1 1
11
0 4 3 0 8 3 2 8 0 7 4
16
0 0 1 1 2 2 3 4 4 5 5 7 8 9 9 10
6
0 0 1 2 3 4
```

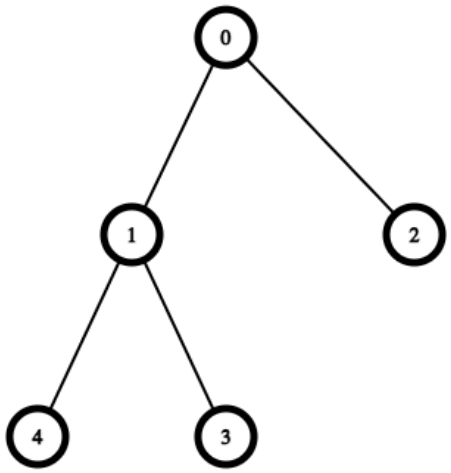
Main Sample Output

```
7
34
41
5
```

Main Sample Explanations

For Test Case #1:

The company structure looks like this:



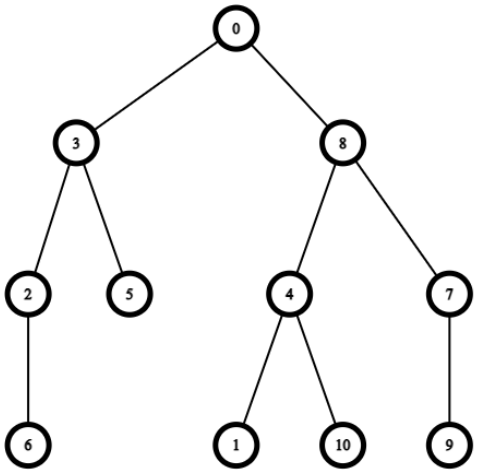
The list of all possible disputes are between:

- 0 and 1: resolved by 0
- 0 and 2: resolved by 0
- 0 and 3: resolved by 0
- 0 and 4: resolved by 0
- 1 and 2: resolved by 0
- 1 and 3: resolved by 1
- 1 and 4: resolved by 1
- 2 and 3: resolved by 0
- 2 and 4: resolved by 0
- 3 and 4: resolved by 1

The maximum number of disputes a single employee is responsible for is 7 (employee 0).

For Test Case #2:

The company structure looks like this:



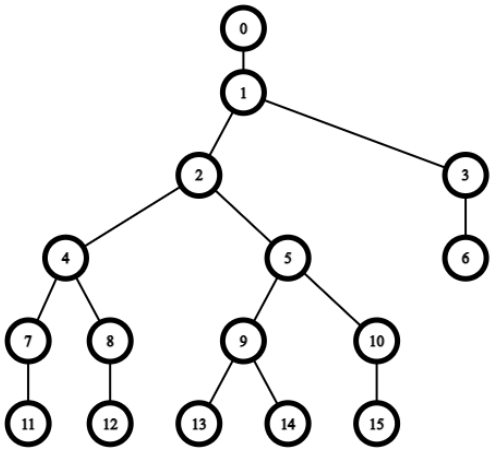
The number of unique disputes each employee is responsible for is as follows:

- Employee 0: 34 disputes
- Employee 1: 0 disputes
- Employee 2: 1 dispute
- Employee 3: 5 disputes
- Employee 4: 3 disputes
- Employee 5: 0 disputes
- Employee 6: 0 disputes
- Employee 7: 1 dispute
- Employee 8: 11 disputes
- Employee 9: 0 disputes
- Employee 10: 0 disputes

The maximum number of disputes a single employee is responsible for is 34 (employee 0).

For Test Case #3:

The company structure looks like this:



The top five employees responsible for the most disputes are:

1. Employee 2 (41 disputes)
2. Employee 1 (38 disputes)
3. Employee 0 (15 disputes)
4. Employee 5 (11 disputes)
5. Employee 4 (8 disputes)

For Test Case #4:

The company structure looks like this:

The top five employees responsible for the most disputes are:

1. Employee 0 (5 disputes)
2. Employee 1 (4 disputes)
3. Employee 2 (3 disputes)
4. Employee 3 (2 disputes)
5. Employee 4 (1 dispute)

Bonus Set 2 Sample Input

```
2
14
0 0 0 0 0 1 1 1 2 3 5 5 5 7
30
0 4 3 0 8 3 2 8 0 7 4 3 3 3 8 7 14 14 8 9 14 10 18 22 22 8 25 26 27 28
```

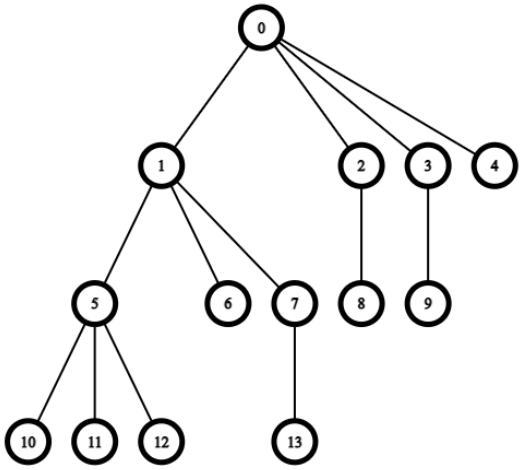
Bonus Set 2 Sample Output

```
61
180
```

Bonus Set 2 Sample Explanations

For Test Case #1:

The company structure looks like this:

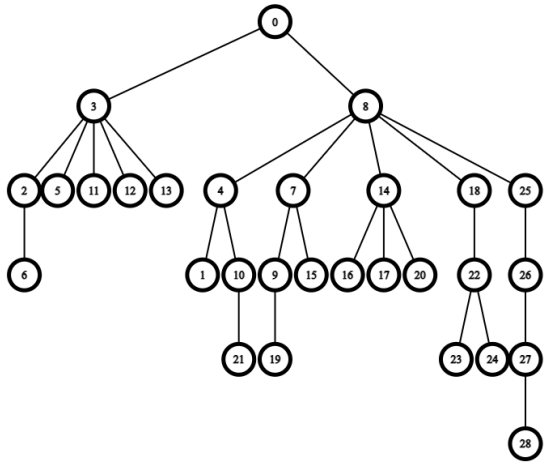


The top three employees responsible for the most disputes are:

1. Employee 0 (61 disputes)
2. Employee 1 (21 disputes)
3. Employee 5 (6 disputes)

For Test Case #2:

The company structure looks like this:



The top four employees responsible for the most disputes are:

1. Employee 8 (180 disputes)
2. Employee 0 (175 disputes)
3. Employee 3 (20 disputes)
4. Employee 14 (6 disputes)