

# FPC 2014 problem presentation; spoiler alert!

Administration  
Back and Forth  
Cryptography  
Diagnosis  
Efficient Pinning  
Friends  
Gardening  
High Towers  
Inaccurate  
Expectations



# A - Administration (1/2)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

Bookkeeping problem: Read in a log file and print the costs for all users if the log is not CORRUPT. Sort the users in the output on alphabetical order (abc...z).

## Solution - Variables:

- `class User(String name, List books, int pay)`
- `TreeMap<String, User> allUsers`
- `Stack<String> bookpile`
- `boolean corrupt`

# A - Administration (2/2)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Solution - Processing

- borrow book: is this book available?
- return book: can this user return this book? How much does he need to pay.
- make books available: check the size of the pile
- don't forget to charge users €10.00 for every book they didn't return.

# Back and Forth

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

- Given a string  $s$  with length  $1 \leq |s| \leq 10^6$ , is this string a palindrome?

## Solution

- Just loop over the string and compare the chars at the beginning with their corresponding places at the end.
- Note that you only have to check the first half of the string, if you didn't; no problem  $10^6$  steps is still acceptable.
- Optionally you could use a `StringBuilder` to reverse the string and match it against  $s$  using the `matches` function.

# Cryptography (1/3)

Administration

Back and Forth

**Cryptography**

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

- Given a number  $n$   $1 \leq n \leq 10^{10}$ , decide whether it's a prime number or not.

## Things to notice

- Since  $n$  can be 10 billion you have to use longs, not integers as they can only store up to 2.1 billion.
- The problem becomes a lot more easy if you know the modulo (%) operator.



# Cryptography (2/3)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Naive approach

- if  $n < 2$  output BROKEN
- else if  $n == 2$  output SAFE
- else loop from  $i = 2$  to  $n$  and check if a number  $i \% n == 0$ . If true output BROKEN else output SAFE.
- This takes approximately  $10^{10}$  steps which would result in TIME LIMIT EXCEEDED.

## First optimization

- Notice that after  $n/2$  no divisor can be found anymore, so loop from 2 to  $n/2$ . This reduces the number of steps to approximately 5 billion, which is unfortunately still too much.



# Cryptography (3/3)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Correct approach

- The correct approach is to loop until the square root of  $n$ .
- You are looking for pairs of numbers  $a$  and  $b$  so that  $a * b = n$  if  $n$  happens to be a composite number. You would only need the smallest of the two and this number must be smaller or equal to  $\sqrt{n}$ , if this would not be the case both  $a$  and  $b$  would be strictly greater than  $\sqrt{n}$  contradicting the fact that  $a * b = n$ .
- Using this approach you end up with approximately  $\sqrt{10^{10}} = 10^5 = 100.000$  which is perfectly fine.
- An optional optimization is to check if  $n \% 2 == 0$  and if not loop from  $i=3$  to  $\sqrt{n}$  where you skip all even numbers by incrementing  $i$  with 2 every time. This would leave you with approximately 50.000 steps.

# Diagnosis

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

Union of all of the sets of symptoms of selected diseases.  
Print "yes" if the diseases clarify all symptoms, no otherwise.

## Solution

- `Set<Integer> symptoms`
- loop over all sets and do `'output.add'` or `'addAll'`
- output contains all symptoms?



# Efficient Pinning

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

- Given two rectangles representing *Board* and *CPU*, count the number of possible ways *CPU* matches subrectangles of *Board*.

## Solution

- Check every possible subrectangle of *Board*, with size equal to *CPU*.
- Count subrectangles that are equal.
- Print the number of matches found.
- Optimization: stop checking a subrectangle as soon as a mismatching *pin* is found.

# Friends

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

**Friends**

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

- Given an graph, check whether all vertices are reachable.

## Solution

- Do a BFS, starting from node  $s$ . Keep track of the visited nodes.
- Check whether all nodes have been visited, print "yes" if so, and "no" if not.

# Gardening (1/2)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

**Gardening**

High Towers

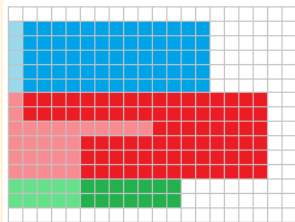
Inaccurate  
Expectations

## Problem description

- Given a sequence of points defining a perimeter, find the area within this perimeter.

## Smart solution

- Calculate the sum of the areas between each line segment and the y-axis.
- $|\sum x_i \cdot (y_i - y_{i+1})|$
- Area is positive if  $y_i > y_{i+1}$  (going down) and negative if  $y_i < y_{i+1}$  (going up).
- Area between tiles and y-axis is added and subtracted, leaving only the total area of tiles.



# Gardening (2/2)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Intuitive solution

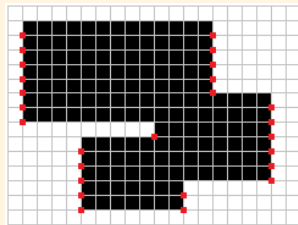
- Use a map to save a list of points for every line.
- Be careful not to count area below or above an edge.
- Going down:

*for*( $j = y_i - 1 \dots y_{i+1}$ )  
*map.get*( $j$ ).*add*( $x$ )

- Going up:

*for*( $j = y_i \dots y_{i+1} - 1$ )  
*map.get*( $j$ ).*add*( $x$ )

- Finally, calculate the total area by adding  $x_{i+1} - x_i$  for every even  $i$  for each row.



# High Towers (1/3)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

- Count the number of upward triangles in a triangle of height  $n$ .

## Recursive formula

- Recursive formula:  $f(n) = f(n-1) + \sum_{i=0}^n i$
- However, with  $n \leq 200000$ , this would result in a stack overflow.

# High Towers (2/3)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Iterative formula

- $f(n) = \sum_{i=0}^n i + \sum_{i=0}^n (i-1) + \dots + \sum_{i=0}^n (i-n) =$
- $\sum_{i=0}^n \frac{i(i+1)}{2} = \frac{1}{2} \sum_{i=0}^n i^2 + i = \frac{1}{2} \left( \sum_{i=0}^n i^2 + \sum_{i=0}^n i \right)$
- Should give correct answer.

## Faster solution

- Direct formula is possible.

# High Towers (3/3)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Things to notice

- $\sum_{i=0}^n i = \frac{1}{2}n(n-1)$
- $\sum_{i=0}^n i^2 = \frac{1}{6}n(n+1)(2n+1)$

## Direct formula

- $f(n) = \frac{1}{12}n(n+1)(2n+1) + \frac{1}{4}n(n+1)$

# Inaccurate Expectations (1/2)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

- For a given  $n$ , output  $g(n)$

## Solution

- "Simply" return  $n + n * g(n - 1)$ , except for  $n = 0$ , which should return 0.
- Expectation:  
 $g(1000) = 109380\ldots[\text{some more digits}]\ldots 20000$ .



# Inaccurate Expectations (2/2)

Administration

Back and Forth

Cryptography

Diagnosis

Efficient Pinning

Friends

Gardening

High Towers

Inaccurate  
Expectations

## Problem description

- For a given  $n$ , output  $g(n)$

## Solution

- "Simply" return  $n + n * g(n - 1)$ , except for  $n = 0$ , which should return 0.
- Expectation:  
 $g(1000) = 109380...[2558 \text{ more digits}]...20000.$
- Use BigInteger