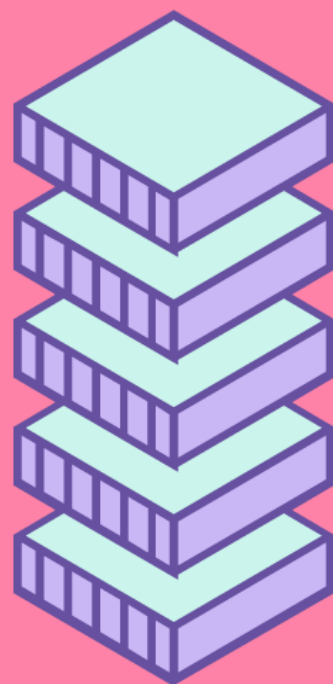




CONTEST

UPSOLVING

PILAS



Ft. Elisa



Recursion: Fibonacci Numbers

Problema visto en clase

El código:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int fibonacci(int n) {
6
7     if(n == 0) return 0;
8     if(n == 1) return 1;
9
10    return fibonacci(n-1)+fibonacci(n-2);
11 }
12
13 int main() {
14     int n;
15     cin >> n;
16     cout << fibonacci(n);
17     return 0;
18 }
```

Minimum Height Triangle

Sea la formula para calcular
el area de un triangulo: $\frac{bh}{2}$
Dado que conocemos el area y la base
podemos "despejar" esas variables.
Y tenemos la siguiente ecuacion: $h = \frac{2A}{b}$

NOTA: Tener cuidado con los "cast"
y con la función round, ceil y floor.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int lowestTriangle(int base, int area){
6     return ceil( (2*area) / (double)base );
7 }
8
9 int main() {
10     int base;
11     int area;
12     cin >> base >> area;
13     int height = lowestTriangle(base, area);
14     cout << height << endl;
15     return 0;
16 }
```

STACKEZ - Easy Stack

Uso básico de las funciones la pila.

Implementación propia.
STL de C++

Solo un detalle importante a
considerar:

La entrada.

Había que usar la fast I/O de C++

```
ios::sync_with_stdio(false);  
cin.tie(NULL); cout.tie(NULL);
```

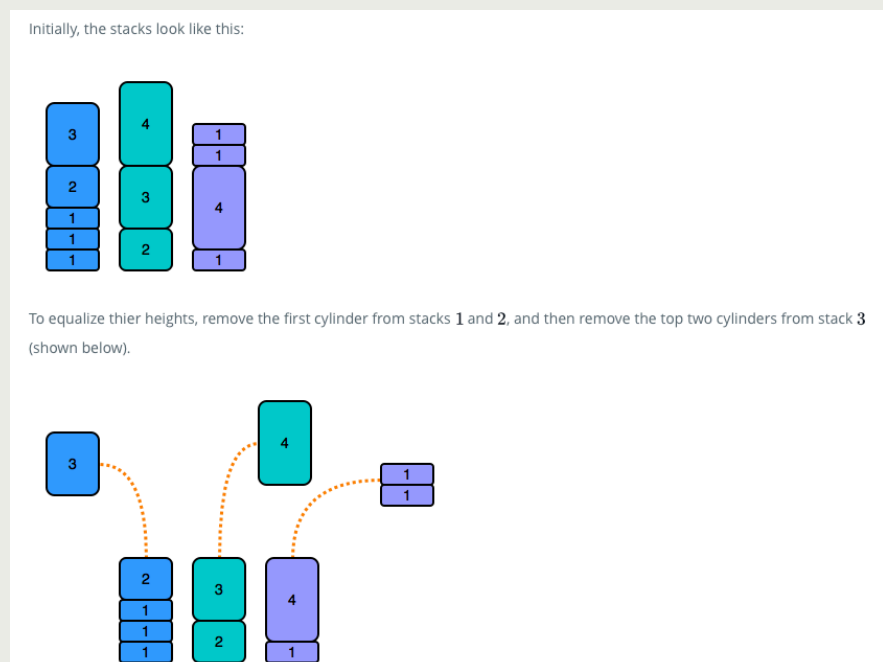
Equal Stacks

Idea de solución

Tenemos nuestras 3 pilas.

Iteramos por las pilas, hasta que
todas las pilas sean iguales o ya no
tengan elementos que tomar.

Y vamos tomando el elemento de la
pila que tenga la suma más grande.



MINSTACK - Smallest on the Stack

Para este problema tenemos que implementar una función que encuentre el mínimo dentro de la pila.

Tenemos muchas formas de resolver este problema.

- Usando un stack de pair
- Usando un stack y un map

Y de la manera más simple, es notando que el problema nunca nos pide el tope de la pila.

Por lo que guardamos el elemento mínimo como tope de la pila y lo vamos haciendo cada que llegue un push.

NOTA: Tener cuidado con el POP y MIN cuando la pila esta vacía.

TMT Document

Este problema tiene muchas soluciones posibles. Y aunque no lo parece, tiene una solución usando una pila.

Idea de solución

Vamos a llevar una pila, donde vamos a hacerle push cada vez que venga un carácter 'T' y cuando exista un carácter 'M' vamos a hacerle pop a la pila, siempre y cuando existan elementos en la pila.

Considerar:

- Tamaño de la pila no debe superar $N/3$. Donde N es el numero de caracteres.
- Contar cada vez que se hace pop a la pila.
- Que el número de pop, al final, sea el mismo que el tamaño de la pila.

Mirror, Mirror

Problema de implementación pura

Lo importante con estos problemas es estar dispuesto a pasar horas, pero al final, si perseveras lo suficiente, veras un AC (:

Cd and pwd commands

El truco de este problema es usar el comportamiento de una pila sobre un arreglo/vector para guardar cada una de las rutas.

Donde cada nodo representa que estamos una carpeta.

Solo tener cuidado cuando la ruta empiece con “/” o cuando sea de tipo “..”.

De tipo, empezando por “/”, borramos el contenido del vector.

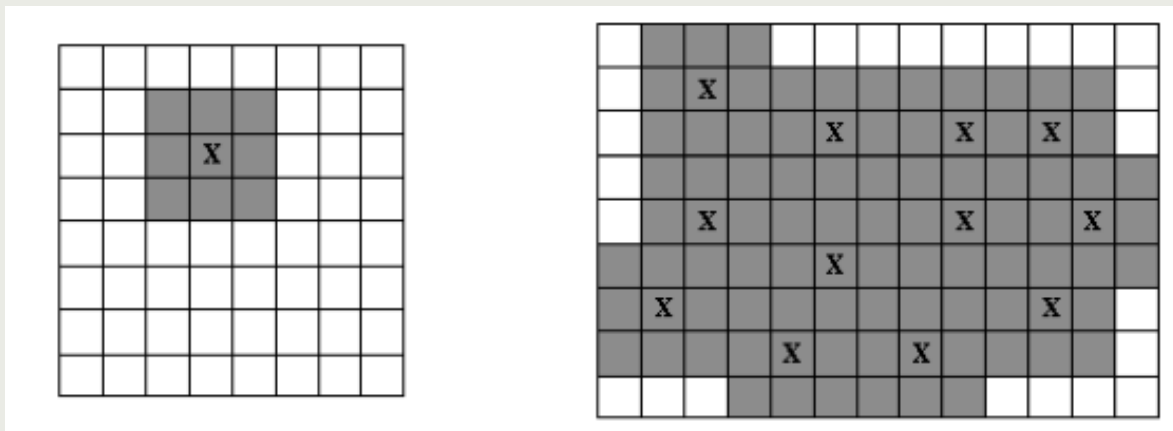
De tipo “..” borramos el “tope” del vector, teniendo cuidado cuando esta vacío.

Tener cuidado al trabajar con las rutas y al concatenar strings.

Searching for Nessy

En el caso de Nessy tenemos que encontrar el mínimo número de sonares para cubrir toda el área de búsqueda en una matriz de $n \times m$. Un punto importante es que los bordes no necesitan ser cubiertos por los sonares.

Notemos que, por lo que nos dice el problema, el sonar cubre una matriz de 3×3 .



Por lo que la solución se traduce en encontrar el mínimo número de sonares para cubrir el ancho y largo de la matriz, pero como sabemos que el sonar cubre 3 cuadritos de ancho y 3 de largo. Que resulta de dividir n entre 3 y m entre 3 y multiplicar el resultado de ambas operaciones.

Links de los problemas

- <https://www.hackerrank.com/challenges/ctci-fibonacci-numbers/problem>
- <https://www.hackerrank.com/contests/infinity18/challenges/lowest-triangle/problem>
- <https://www.spoj.com/problems/STACKEZ/>
- <https://www.hackerrank.com/contests/june-world-codesprint/challenges/equal-stacks/problem>
- <https://www.spoj.com/problems/MINSTACK/>
- <https://codeforces.com/problemset/problem/1509/B>
- https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=407
- <https://codeforces.com/problemset/problem/158/C>
- https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=1985

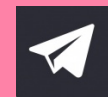
Contacto:



gomezerm@gmail.com



[@elissabjg](#)



[@elissabj](#)