CS2023 - Aula de Ejercicios Nº 4

Heider Sanchez
ACL: Juan Diego Castro
Semestre 2024-1

- Cada estudiante debe resolver los ejercicios de forma **individual**.
- Cualquier indicio de plagio será calificado con 0.
- Entregar los archivos **ejercicio1.cpp**, **ejercicio2.cpp** y **ejercicio3.cpp** sin comprimir. Cualquier otro archivo **no será revisado**.

Ejercicios

1. (7 pts) Dado un string con paréntesis balanceados s, retornar el **score** del string. El **score** de un string con paréntesis balanceados se determina de la siguiente forma:

   - "()" tiene score 1
   - AB tiene score A+B, donde A y B son 2 strings con paréntesis balanceados
   - (A) tiene score 2*A, donde A es un string con paréntesis balanceados

   - **Ejemplo 1:**
     Input: s = "()"
     Output: 1
   - **Ejemplo 2:**
     Input: s = "(())"
     Output: 2
   - **Ejemplo 3:**
     Input: s = "()()"
     Output: 2

   Restricciones:

   - $2 \leq s.length \leq 50$
   - s sólo contiene "(" y ")"
   - s está balanceado

2. (7 pts) Implementa una queue (FIFO) usando únicamente 2 stacks. La queue debe ser capaz de soportar las operaciones básicas de push, peek, pop y empty.
   Implementa la clase MyQueue:

```
class MyQueue {
public:
    MyQueue() {}
    void push(int x) {}
    int pop() {}
    int peek() {}
    bool empty() {}
};
```

   - void push(int x) Pone el elemento x al final de la cola
   - int pop() Remueve el elemento del frente de la cola y lo retorna.

- int peek() Retorna el elemento al frente de la cola.
- boolean empty() Retorna true si la cola está vacía y false en caso contrario.

Ejemplo :

Inputs:

$[$"$MyQueue$","$push$","$push$","$peek$","pop","$empty$"$]$

$[[], [1], [2], [], [], []]$

Output: $[null, null, null, 1, 1, false]$

Explicación

MyQueue myQueue = new MyQueue();

myQueue.push(1); // queue: [1]

myQueue.push(2); // queue: [1, 2] (el más a la izquierda es el front de la cola)

myQueue.peek(); // retorna 1

myQueue.pop(); // retorna 1 y queue es [2]

myQueue.empty(); // retorna false

Restricciones:

- $1 \leq x \leq 9$
- Todas las llamadas a pop y peek son válidas

3. (6 pts) Dado un string s que contiene '(', ')', ' {', '}', '[' y ']'. Determine si s es válido.
   Se dice que s es válido si:
   - Los brackets abiertos son cerrados con el mismo tipo de brackets
   - Los brackets abiertos son cerrados en el orden correcto
   - Cada bracket cerrado tiene un bracket abierto correspondiente y es del mismo tipo

   Ejemplos:

   - Ejemplo 1:
     Input: $s =$ "({})"
     Output: true
   - Ejemplo 2:
     Input: s = "()[]{}"
     Output: true
   - Ejemplo 3:
     Input: $s =$ "(]"
     Output: false

   Restricciones:

   - $1 \leq s.length \leq 104$
   - s sólo contiene brackets: '()[]{}'