Taller 2: Pilas (fundamentos)

EDyA 361075 Carlos Eduardo Sánchez Torres

20 de febrero de 2021

1. Descripción de la actividad

Crear un programa para el manejo de pilas.

1. Introducir datos a la pila.
2. Visualizar los datos en la pila.
3. Sacar los datos de la pila.
4. Visualizar la salida de los datos de la pila.

2. Resultados

Un conjunto fundamental dinámico y finito en ciencias computacional, operaciones básicas necesarias para dichos conjuntos son búsqueda, inserción, eliminación, mínimo, máximo, sucesor y predecesor. En una pila, el elemento eliminado desde el el conjunto es el más reciente insertado: la pila implementa una política último en entrar y primero en salir. Los beneficios de dicha de estructura se muestran al insertar elementos $\mathcal{O}(1)$ y remover el último $\mathcal{O}(1)$. Pero presenta problemas al intentar las mismas operaciones en el medio del conjunto.

Listing 1: La pila es vacía. $\mathcal{O}(1)$

```
bool Stack::isEmpty() {
    return this->length == 0;
}
```

Listing 2: Añade un nuevo nodo. $\mathcal{O}(1)$

```
void Stack::push(T element) {
    this->top = new Node<T>(element, this->top);
    this->length++;
}
```

Listing 3: Saca el último nodo insertado. $\mathcal{O}(1)$

```
T Stack::pop() {
    if (this->isEmpty()) {
        return 0;
    }
    this->length--;
    return Node<T>::moveToNext(*this->top);
}
```

3. Anexos

El resto del código puede ser encontrado en mi GitHub [1].

```cpp
#include "DataStructure.h"
#include "iostream"
using namespace std;
using namespace DataStructure;

int main() {
    Stack<int> stack;
    // 1
    stack.push( element: 1);
    // 10 1
    stack.push( element: 10);
    // 100 10 1
    stack.push( element: 100);
    // 100
    cout << stack.pop() << "\n";
    // 10
    cout << stack.pop() << "\n";
    // 1
    cout << stack.pop() << "\n";
    return 0;
}
```

Figura 1: Main.

```
#include "Node.h"
#ifndef UABC_STACK_H
#define UABC_STACK_H

template<class T>
class Stack {
private:
    int length = 0;
    Node<T>* top = nullptr;
public:
    bool isEmpty();
    void push(T element);
    T peek();
    T pop();
    int size();
};

#endif //UABC_STACK_H
```

Figura 2: Headers.

Referencias

[1] [1] sanchezcarlosjr. 2021. sanchezcarlosjr/uabc. GitHub. Retrieved February 20, 2021 from https://github.com/sanchezcarlosjr/uabc/tree/master/src/introduction-to-algorithms/week3-practice1

[2] Cormen, T. H., &; Leiserson, C. E. (2009). Introduction to Algorithms, 3rd edition. In Introduction to algorithms, 3rd edition (pp. 232-233).