

TALLER PROGCOMP: TRACK BÁSICO

ESTRUCTURAS DE DATOS NO LINEALES III

Gabriel Carmona Tabja

Universidad Técnica Federico Santa María,
Università di Pisa

June 10, 2024

Part I

EDD NO LINEALES III

EDD No LINEALES

Aprendidas

- ▶ Diccionarios (map C++)
- ▶ Conjunto (set C++)
- ▶ Multi Conjunto (multiset C++)

La que vamos a aprender

- ▶ Cola de Prioridad

Part II

COLA DE PRIORIDAD

COLA DE PRIORIDAD

Definición

- ▶ Permite identificar el mayor/menor elemento de un grupo de elementos en $O(1)$

COLA DE PRIORIDAD

Definición

- ▶ Permite identificar el mayor/menor elemento de un grupo de elementos en $O(1)$
- ▶ Permite inserción y borrado en $O(\log n)$

COLA DE PRIORIDAD

Definición

- ▶ Permite identificar el mayor/menor elemento de un grupo de elementos en $O(1)$
- ▶ Permite inserción y borrado en $O(\log n)$
- ▶ Permite repetidos

COLA DE PRIORIDAD

Definición

- ▶ Permite identificar el mayor/menor elemento de un grupo de elementos en $O(1)$
- ▶ Permite inserción y borrado en $O(\log n)$
- ▶ Permite repetidos

Implementación

- ▶ Árbol binario casi-completo, conocido como heap

COLA DE PRIORIDAD

Definición

- ▶ Permite identificar el mayor/menor elemento de un grupo de elementos en $O(1)$
- ▶ Permite inserción y borrado en $O(\log n)$
- ▶ Permite repetidos

Implementación

- ▶ Árbol binario casi-completo, conocido como heap
- ▶ Solo basta un arreglo

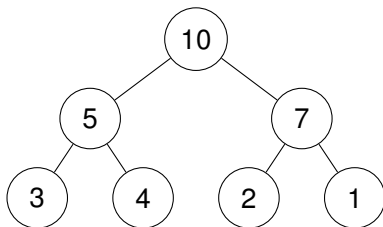
COLA DE PRIORIDAD

Definición

- ▶ Permite identificar el mayor/menor elemento de un grupo de elementos en $O(1)$
- ▶ Permite inserción y borrado en $O(\log n)$
- ▶ Permite repetidos

Implementación

- ▶ Árbol binario casi-completo, conocido como heap
- ▶ Solo basta un arreglo



heap		10	5	7	3	4	2	1
pos	0	1	2	3	4	5	6	7

UTILIZACIÓN EN C++ - BÁSICO

```
1  int main() {
2      priority_queue< int > pq; // acceso a mayor O(1)
3
4      pq.push(2);
5      pq.push(4);
6      pq.push(3);
7      pq.push(7);
8      pq.push(4);
9      pq.push(2);
10
11     cout << pq.size() << "\n";
12     cout << pq.top() << "\n";
13     while(!pq.empty()) {
14         cout << pq.top() << "\n";
15         pq.pop();
16     }
17 }
```

UTILIZACIÓN EN C++ - MODIFICABLE

```
1  int main() {
2      priority_queue< int, vector< int >, greater< int > > pq; // acceso a menor O(1)
3
4      pq.push(2);
5      pq.push(4);
6      pq.push(3);
7      pq.push(7);
8      pq.push(4);
9      pq.push(2);
10
11     cout << pq.size() << "\n";
12     cout << pq.top() << "\n";
13     while(!pq.empty()) {
14         cout << pq.top() << "\n";
15         pq.pop();
16     }
17 }
```

REFERENCES I