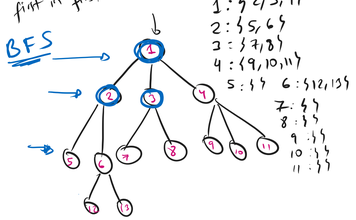
Queue (cola)

FIFO
first in first out

BFS



1: {2,3,4}
2: {5,6}
3: {7,8}
4: {9,10,11}
5: {} 6: {12,13}
7: {}
8: {}
9: {}
10: {}
11: {}

queue<int> q;

q.push(1): q={1}

int x = q.front(); q.pop() q={}

1 → insert 2,3,4 → q={2,3,4}

x = q.front(); q.pop()

2 → insert 5,6 → q={4,5,6}

x = q.front() q.pop()

3 → insert 7,8 → q={4,5,6,7,8}

STACK (LIFO)

last in first out

stack<int> s 2, 3, 1, 5

s.top()

Uso: Regular Bracket

(( )) ( ) pila vacía

open=0

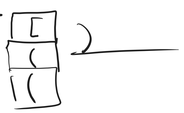
1 2 1 0 1 0

* Al final open = 0
* open ≥ 0 en todo el proceso

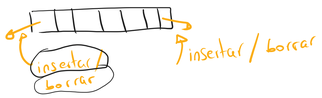
(({ } [( ) ( )] { }))

* Saber el último carácter
* Eliminar el último

Pila



Deque



K=3

	1	2	3	4	5	6	7
wins →	3	5	1	2	6	4	7
	0	0	0	0	0	0	0

↑ K

wins(x) == K

pq: 1 3 8 6

↳ árbol: heap

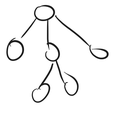
8 6 3 1
-1 -3 -6 -8

max-heap — min-heap

insert: log N
log N
elim

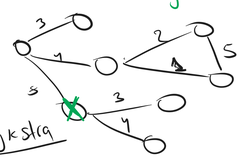
MAX-PR. O(1)

MIN-PR



priority_queue<T, vector<T>, comp>

greater<T>



Dijkstra

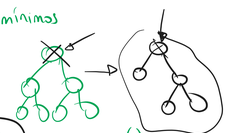
↳ pq

* Extraer 2 mínimos
* Insertar

k=3

min

pq



int x = pq.top(); pq.pop();

int y = pq.top(); pq.pop();

int z = x + 2y

pq.push(z) if(x ≥ k) break

if(pq.size() == 1) break

Ins 2
Ins 3
Ins 7
Ins 4
Ins 1

multiset.erase(-1)

- vector
- stack
- queue
- priority queue
- map
- set
- multimap
- multiset
- deque