STL

In this lesson, we'll see how to use the STL priority queue as a built-in heap.

We'll cover the following

Library

While solving a problem, we’ll be using heap from C++ STL (`priority_queue`).

Below are some of the operations that we’ll use frequently. For complete documentation about how they work, check here.

> By default, in C++, `priority_queue` is a **max heap**.

```
priority_queue<int> Q; //empty heap
int x = S.top(); // get top element
Q.pop(); // pop top element (maximum element)
Q.push(x); //add x to the heap
bool x = Q.empty(); // to check if heap is empty
int sz = Q.size(); // get size of heap
```

Min heap

There are two ways to use `priority_queue` as a min-heap.

Insert negative elements

Min heap for `[2, -3, 4, -1]` is just Max heap for `[-2, 3, -4, 1]`.

Use three arguments constructor

`priority_queue<T>` has three argument overloaded constructor.

`priority_queue <Type, Container, Compare>`. The default values are

- Container: `vector<Type>`
- Compare: `less<Type>`

Below is the declaration for min-heap:

`priority_queue <int, vector<int>, greater<int> > Q;`

In the next lesson, we’ll discuss a solved problem using heaps.