Detail

Heap in C++ STL

(Training materials for students)

Heap data structure can be implemented in a range using STL which allows faster input into heap and retrieval of a number always results in the largest number i.e. largest number of the remaining numbers is popped out each time. Other numbers of the heap are arranged depending upon the implementation.

Operations on heap:

- **make_heap**() :- This function is used to convert a range in a container to a heap.
- **front**() :- This function displays the first element of heap which is the maximum number.
- **push_heap**() :- This function is used to insert elements into heap. The size of the heap is increased by 1. New element is placed appropriately in the heap.
- **pop_heap**() :- This function is used to delete the maximum element of the heap. The size of heap is decreased by 1. The heap elements are reorganised accordingly after this operation.
- **sort_heap**() :- This function is used to sort the heap. After this operation, the container is no longer a heap.
- **is_heap**() :- This function is used to check whether the container is heap or not. Generally, in most implementations, the reverse sorted container is considered as heap. Returns true if container is heap else returns false.
- **is_heap_until**() :- This function returns the iterator to the position till the container is the heap. Generally, in most implementations, the reverse sorted container is considered as heap.

--- The End ---