Time Complexities

1. Array / Vector

Function	Time Complexity	Space Complexity
sort (v.begin(), v.end())	O(nlog(n))	O(logn)
reverse (v.begin(), v.end())	O(n)	O(1)
v.push_back (x)	O(1)	O(1)
v.pop_back ()	O(1)	O(1)
v.size()	O(1)	O(1)
v.clear()	O(n)	O(1)
v.erase()	O(n^2)	O(1)

2. deque

i) The time complexities of all deque operation O(1), except random acess and inserting in middle whose complexity is O(n).

3. List

All list operations have O(1) time complexity except assinging value whose is O(n).

4. Stack

function	Time Complexity	Space Complexity
s.top()	O(1)	O(1)
s.pop()	O(1)	O(1)
s.empty()	O(1)	O(1)
s.push(x)	O(1)	O(1)

5. Queue

Function	Time Complexity	Space Complexity
q.push(x)	O(1)	O(1)
q.pop()	O(1)	O(1)
q.front()	O(1)	O(1)
q.back()	O(1)	O(1)
q.empty()	O(1)	O(1)
q.size()	O(1)	O(1)

6. Priority Queue

Q.top()	O(1)	O(1)
Q.push()	O(log n)	O(1)
Q.pop()	O(log n)	O(1)
Q.empty()	O(1)	O(1)

7. Set

s.find()	O(log n)	O(1)
s.insert(x)	O(log n)	O(1)
s.erase(x)	O(log n)	O(1)
s.size()	O(1)	O(1)
s.empty()	O(1)	O(1)

Notes ⟹ Unordered_set have find() time complexity O(1) and rest are same.

8. Map

Function	Time Complexity	Space Complexity
m.find(x)	O(log n)	O(1)
m.insert(pair<int,int>(x,y))	O(log n)	O(1)
m.erase(x)	O(log n)	O(1)
m.empty()	O(1)	O(1)
m.clear()	O(n)	O(1)
m.size()	O(1)	O(1)

⟹ Unordered_map have everything same except its search time (find funct^n) is O(1)