

# C++ STL: Policy based data structures

## Operations:

1. insert: same as set.
2. find\_by\_order(): returns an iterator to the k-th largest element (counting from zero).
3. order\_of\_key(): the number of items in a set that are strictly smaller than our item.
4. find: same as set
5. erase: same as set.
6. size: same as set.
7. begin, end, rbegin, rend: same as set.
8. lower\_bound, upper\_bound: same as set.

Important note: A drawback of using less\_equal instead of less is that lower\_bound works as upper\_bound so it's better to use pair instead of int.

```
#include <bits/stdc++.h>

#include <ext/pb_ds/assoc_container.hpp>

using namespace __gnu_pbds;

using namespace std;

typedef long long ll;

typedef tree<
    pair<ll, ll>, //change data type
    null_type,
    less<pair<ll, ll>>, //change comparator and data type, options: [less, less_equal]
    rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

const int N = 1e5 + 9, M = 1e6 + 9, OO = 0x3f3f3f3f;

ll l100 = 0x3f3f3f3f3f3f3f3f;

int main() {
    ordered_set X;

    X.insert(1);
    X.insert(2);
    X.insert(4);
    X.insert(8);
    X.insert(16);

    cout << *X.find_by_order(1) << endl; // 2
    cout << *X.find_by_order(2) << endl; // 4
    cout << *X.find_by_order(4) << endl; // 16
    cout << (X.end() == X.find_by_order(6)) << endl; // true

    cout << X.order_of_key(-5) << endl; // 0
    cout << X.order_of_key(1) << endl; // 0
    cout << X.order_of_key(3) << endl; // 2
    cout << X.order_of_key(4) << endl; // 2
    cout << X.order_of_key(400) << endl; // 5
}
```

I got error importing this -> #include<ext/pb\_ds/assoc\_container.hpp> Error was "cannot open source file hash\_standard\_resize\_policy\_imp.hpp".

Fix go to the dir -> C:\MinGW\lib\gcc\mingw32\8.2.0\include\c++\ext\pb\_ds\detail\resize\_policy

There u will see a file similar to -> "hash\_standard\_resize\_policy\_imp.hpp0000644"

Rename it to hash\_standard\_resize\_policy\_imp.hpp