PAIRS:

Pair<int ,int > p;

p.first=10;

p.second=12;

cout<<p.first<<” “<<p.second;

VECTOR:

Vector<int,int> v;

LIST:

Similar to vector

But gives front operation as well

In list internal operation doubly linkedlist.

In vector internal operation is singly linkedlist.

list<int> ls;

ls.push\_back(10);

ls.emplace\_back(20);

ls.push\_front(15);

//other function same as vector

//begin,end,rend,clear,size,swap

DEQUE:

Same as vector

STACK:

//LIFO

Stack<int> st;

st.push(10);

st.push(15);

st.push(20);

cout<<st.top();//20

st.pop();//pops 20

st.size()

st.empty()

QUEUE:

//FIFO

queue<int> q;

q.push(10);

q.push(20);

cout<<q.front();//10

cout<<q.back();//20

PRORITY QUEUE:

Priority\_queue<int> pq;

Pq.push(10);

Pq.push(20);

Pq.push(30);

Pq will contain //30,20,10;

//minimal heap

Priority\_Queue<int,vector<int>,greater<int>>pq;

Pq.push(10);

Pq.push(20);

Pq.push(30);

Pq will contain//10,20,30

TIME COMPLEXITY:

* PUSH(log(n))
* POP(log(n))
* TOP(O(1))

SET:

//stores everything sorted and unique.//vector//LIFO

Set<int>s;

s.insert(10);

s.insert (10);

s.insert (20):

s.insert (30);

s will contain//10,20,30

auto it=s.find(30);//returns an iterator

auto it=s.finf(40);//if value not there then return address of garbage after last value in the set.

s.erase(10);//erase 10//log(n)//time complexity.

Int cnt=s.count(20);//if there gives 1 or 0;//unique

s.erase(it);//O(1);

s.lower\_bound(10);//give the iterator pointing element if present then ok else the next elemt is printed//10//20

s.upper\_bound(10);//gives the iterator pointing next to the pointing element else same the next element//20//20

MULTISET:

//sorted//not unique//1122333444

Multiset<int>ms;

Ms.insert(10);

Ms.insert(10);

Ms.insert(10);//ms={10,10,10}

Ms.erase(10);//erase all

Ms.erase(ms.find(10));//only erase 10

UNORDERED\_SET:

//all unique//not sorted.

//o(n)//o(1)//worst//best

MAP:

//unique key//sorted

Map<int,int>mpp;

Map<int,pait<int,int>>mpp;//key

//same as set

//log(n)

MULTIMAP:

//not unique key//sorted

UNORDERDMAP:

//similar to unorderd-set//unique//not sorted

//O(n)

SORT

//sort(a,a+n)

//sort(a+2,a+4)

//sort(a,a+n,greater<int>);//sort in desending order;

Pair<int,int> p={{1,2},{2,4},{3,2}};

//sort(a,a+n,comp);

Bool comp (pair<int,int>p1,pair<int,int>p2){

If(p1.second>p2.second)return false;

If(p1.second<p2.second)return true;

If(p1.first>p2. first)return true;

Return false;

}

Builtin popcount:

Int num=7;

Int cnt=\_builtin\_popcount();//for long long//\_builtiin\_popcountll();

//Returns//no of setbits;//for 7 it is 3

Next\_permutation:

//should be sorted for all permuatation

Do{

Cout<<s<<endl;

}while(next\_permutation(s.begin(),s.end()));

Max//min

Int max=\*max\_element(a,a+n);//address of iterator due to star then element

//similarly minimum element