**Бинарный поиск**

pii BS(int L, int R, int X)

{

while( !(L > R) )

{

int M = L+(R-L)/2;

//cout <<L <<" " <<R <<"\n";

if (mass[M] == X) return mp(M, 1);

if (mass[M] < X) L = M+1;

if (mass[M] > X) R = M-1;

}

return mp(L, 0);

}

pii BS\_R(int L, int R, int X)

{

if (L > R) return mp(L, 0);

int M = L+(R-L)/2;

//cout <<L <<" " <<R << " " << M <<"\n";

if (mass[M] == X) return mp(M,1);

if (mass[M] < X) return BS\_R(M+1, R, X);

if (mass[M] > X) return BS\_R(L, M-1, X);

}

int X = 20, Y = 6;

int myints[] = {10,20,30,30,20,10,10,20};

vector<int> v(myints,myints+8);

vector<int>::iterator low,up;

int main ()

{

sort (v.begin(), v.end()); // 10 10 10 20 20 20 30 30

low = lower\_bound(v.begin(), v.end(), X); // ^

up = upper\_bound (v.begin(), v.end(), X); // ^

cout << "lower\_bound at position " << (low- v.begin()) << endl;

cout << "upper\_bound at position " << (up - v.begin()) << endl;

if ( binary\_search (v.begin(), v.end(), Y) ) cout << "found!\n";

else cout << "not found.\n";

return 0;

}