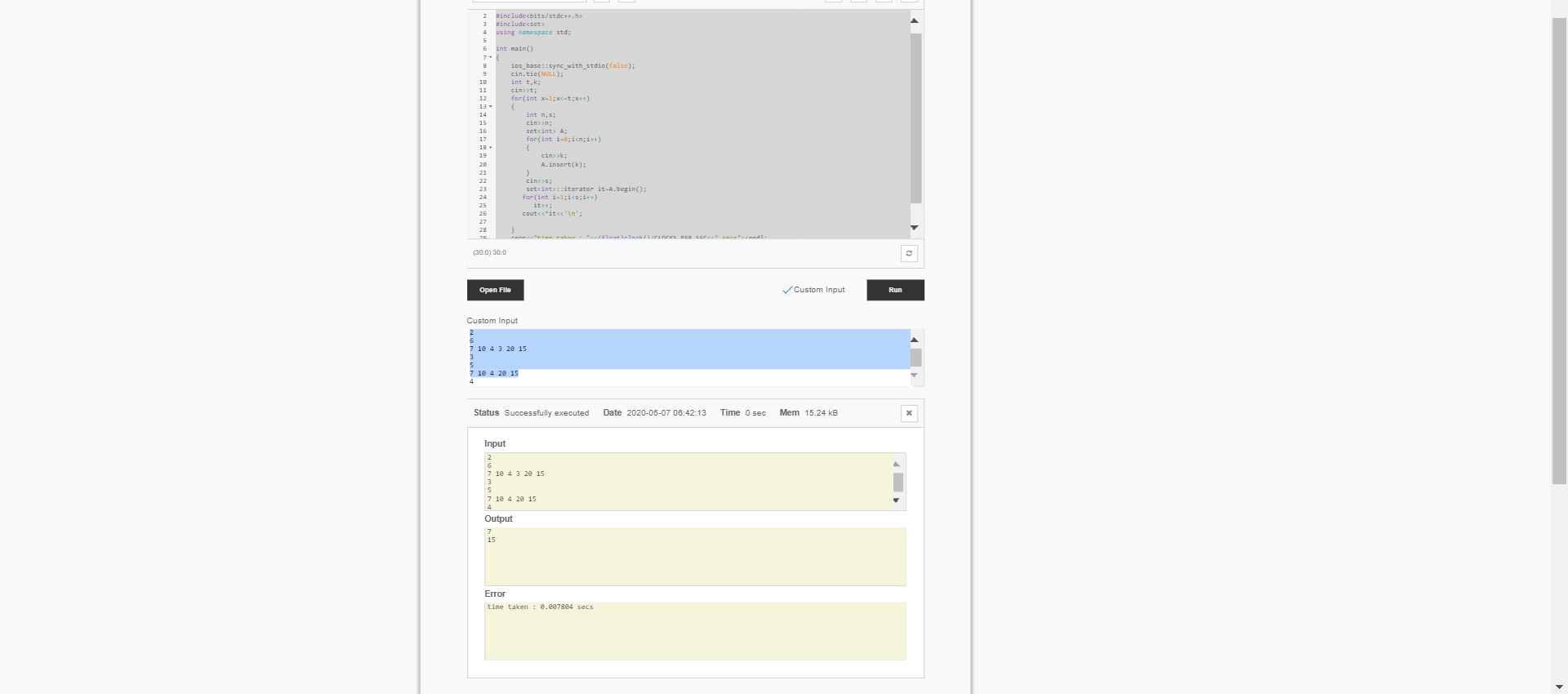
**Hey, I went through your code and I found it was alright. No errors or something of that sort. In case of time limit, I ran your code and found it was running perfectly fine… I am not sure about larger testcases but for sample test cases it was running fine and is expected to run for larger values too.**



But in case you want more efficient approach, there’s a new library called PBDS. Its just like set with many more added features…. This code using PBDS looks like :

#include <bits/stdc++.h>

#include <ext/pb\_ds/assoc\_container.hpp>

#include <ext/pb\_ds/tree\_policy.hpp>

using namespace std;

using namespace \_\_gnu\_pbds;

typedef tree<int, null\_type, less<int>, rb\_tree\_tag, tree\_order\_statistics\_node\_update> pbds;

int main(){

int test;

cin>>test;

while(test--)

{

pbds A;

int n, num;

cin>>n;

for(int i=0; i<n; i++)

{

cin>>num;

A.insert(num);

}

int k;

cin>>k;

cout << k << "th smallest : " << \*A.find\_by\_order(k-1) << endl;

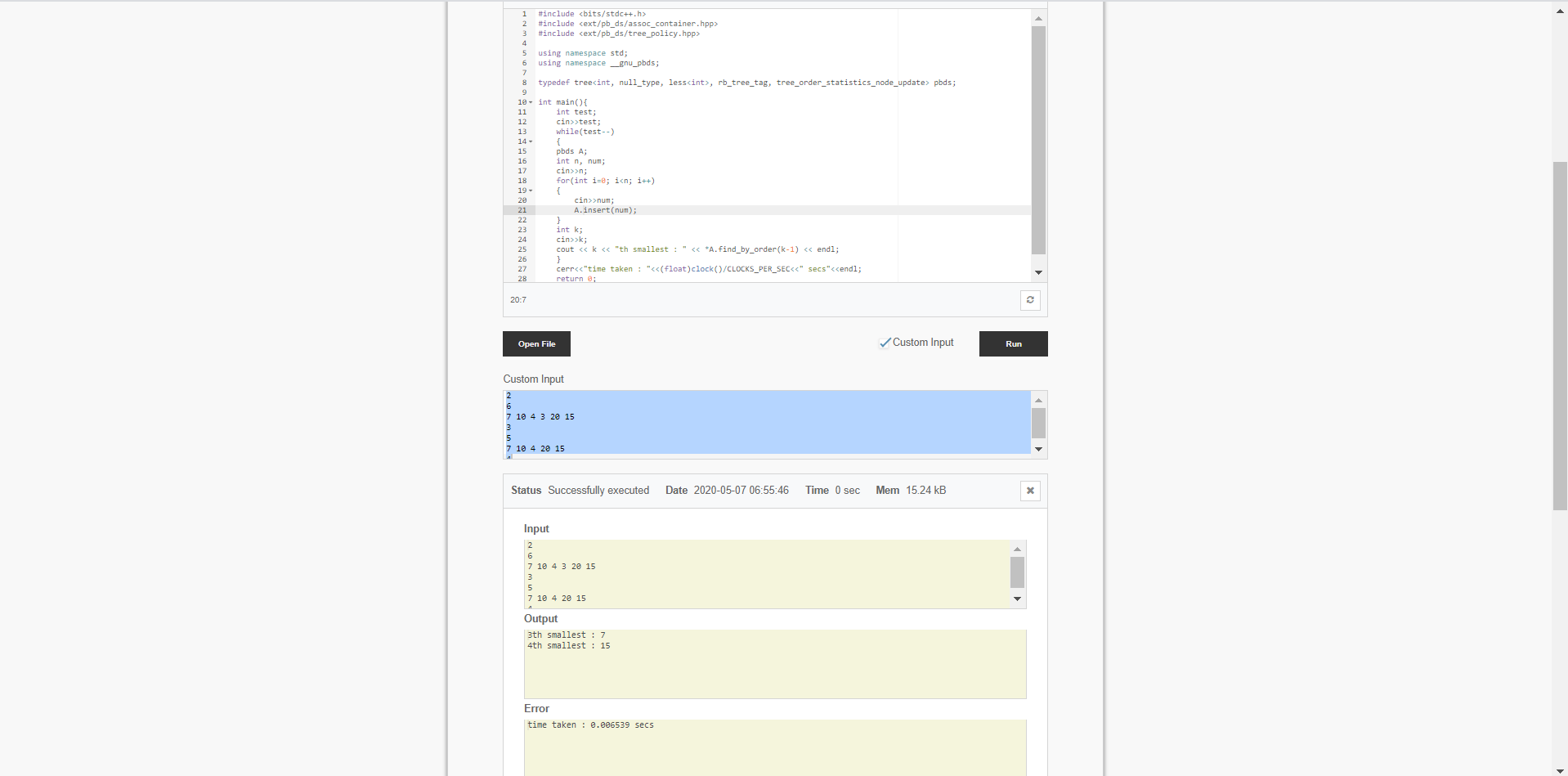
}

cerr<<"time taken : "<<(float)clock()/CLOCKS\_PER\_SEC<<" secs"<<endl;

return 0;

}

And it executes in comparatively lesser time :



Learn more about it from GeeksforGeeks or Tutorials Point. Its efficient in some cases…