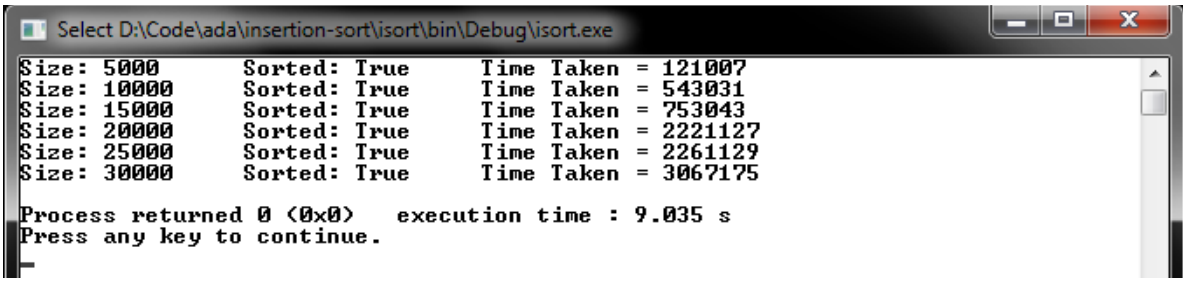
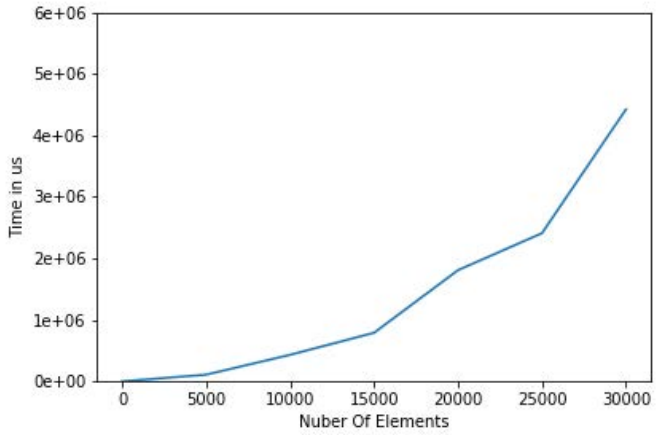
Insertion Sort

```
#include <algorithm>
#include <iostream>
#include <vector>
#include <random>
#include <time.h>
#define LL long long
using namespace std;

void isort(vector<LL>& arr) {
   for(LL i=1; i<arr.size(); i++) {
      if(arr[i] < arr[i-1]) {
         LL temp = arr[i];
         LL j=-1;
         for(j=i-1; j>=0 && temp < arr[j]; j--) {
            arr[j+1] = arr[j];
         }
         arr[j] = temp;
      }
   }
}

mt19937 rnd;
uniform_int_distribution<uint64_t> dist(0,1e4);
LL gen() {
   return dist(rnd);
}

int main()
{
   vector<LL> time;
   for(LL n=1; n<=30001; n+=5000)
   {
      vector<LL> V(n,0);
      //Average Case:
      generate(V.begin(), V.end(), gen);
      //Best Case:
      /*for(LL i=0; i<n; i++) {
         V[i] = i;
      }*/
      //Worst Case: use n-i instrad of i
      starttime = getTime();
      isort(V);
      endtime = getTime();
      time.push_back(endtime-starttime);
      cout<<"Size: "<<n-1<<"\tSorted: "<<(is_sorted(V.begin(), V.end()) == 1? "True\t":"False\t"); printTime(); cout<<endl;
   }
   return 0;
}
```

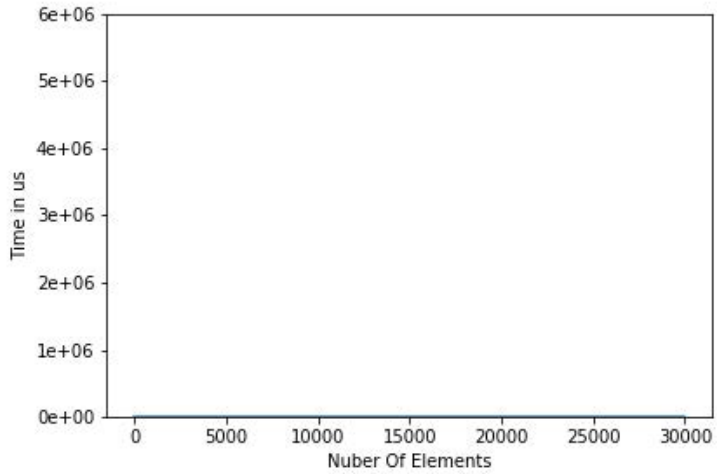
Output:



Average Case:



Best Case:



Worst Case: