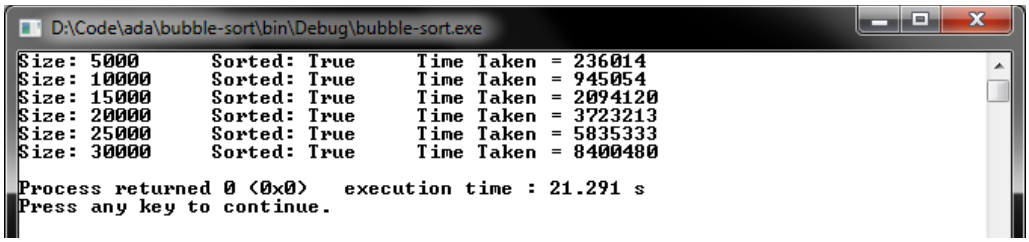
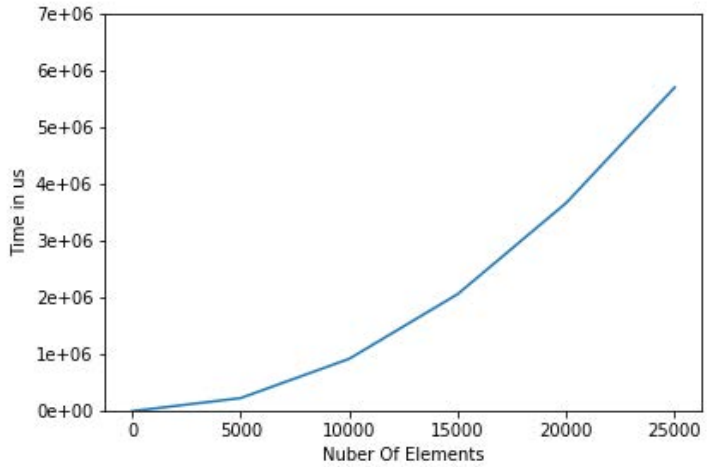
Bubble Sort

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

void bsort(vector<LL>& arr) {
   for(LL i=arr.size()-1; i>=0; i--) {
      for(LL j=0; j<i; j++) {
         if(arr[j] > arr[j+1]) {
            LL temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = 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();
      bsort(V);
      endtime = getTime();
      time.push_back(endtime-starttime);
      cout<<"Sorted: "<<(is_sorted(V.begin(), V.end()) == 1? "True":"False")<<endl;
   }

   return 0;
}
```

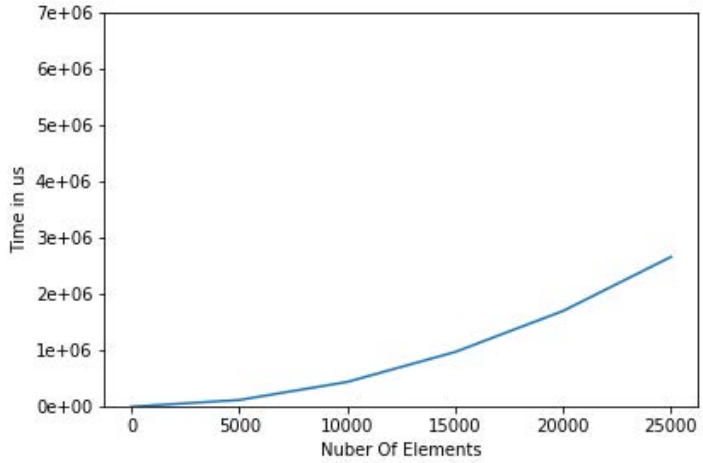
Output:



Average Case:



Best Case:



Worst Case: