Merge Sort

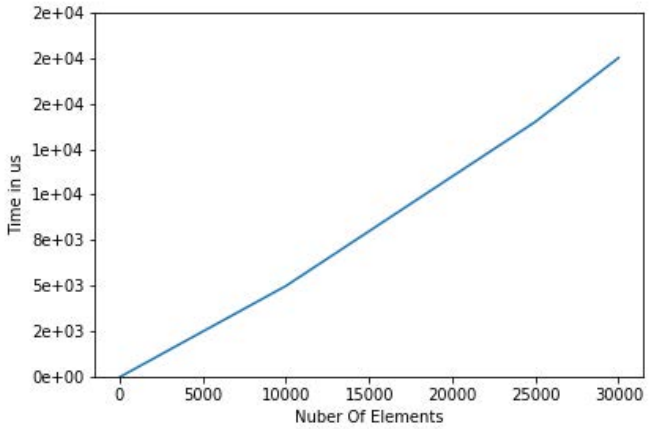
```
#include <algorithm>
#include <iostream>
#include <vector>
#include <random>
#include <time.h>
#define LL long long
using namespace std;

void merge(vector<LL>& arr, vector<LL>& brr, LL start, LL mid, LL end) {
   LL l=start, r=mid, idx=start;
   while(l!=mid && r!=end) {
      if(arr[l]<=arr[r]) {
         brr[idx] = arr[l];
         l++; idx++;
      }
      else if(arr[r]<arr[l]) {
         brr[idx] = arr[r];
         r++; idx++;
      }
   }
   while(l!=mid) {
      brr[idx] = arr[l];
      l++; idx++;
   }
   while(r!=end) {
      brr[idx] = arr[r];
      r++; idx++;
   }
   for(LL i=start; i<end; i++) {
      arr[i] = brr[i];
   }
}

void msort(vector<LL>& arr, vector<LL>& brr, LL start, LL end) {
   if(start >= end) {
      brr[start] = arr[start];
      return;
   }
   LL mid = (start+end)/2;
   msort(arr,brr,start,mid);
   msort(arr,brr,mid+1,end);
   merge(arr,brr,start,mid+1,end+1);
}
```

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

int main()
{
   vector<LL> time;

   for(LL n=5001; n<=30001; n+=5000)
   {
      vector<LL> arr(n,0), brr(n,0);
      //Average Case:
      generate(arr.begin(), arr.end(), gen);
      //Best Case:
      //for(LL i=0; i<n; i++) {
      //    arr[i] = n-i;
      //}
      //Worst Case: use n-i instrad of i
      starttime = getTime();
      msort(arr,brr,0,arr.size()-1);
      endtime = getTime();
      time.push_back(endtime-starttime);
      cout<<"Size: "<<n-1<<"\tSorted: "<<(is_sorted(brr.begin(), brr.end()) == 1? "True\t":"False\t");
      printTime(); cout<<endl;
   }

   return 0;
}
```

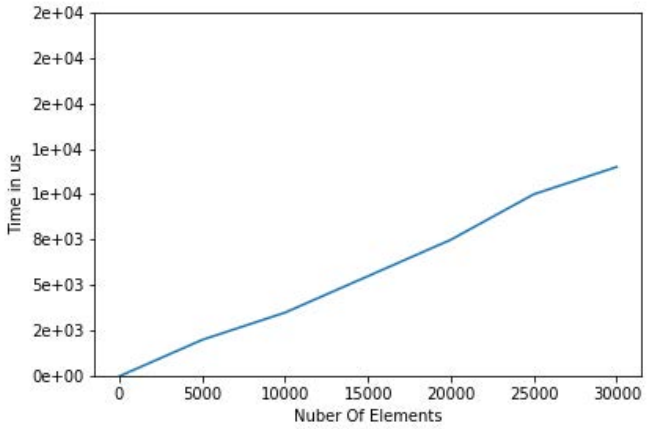
Output:

```
D:\Code\ada\merge-sort\all\bin\Debug\all.exe
Size: 5000      Sorted: True      Time Taken = 2000
Size: 10000     Sorted: True      Time Taken = 5000
Size: 15000     Sorted: True      Time Taken = 8000
Size: 20000     Sorted: True      Time Taken = 11001
Size: 25000     Sorted: True      Time Taken = 13001
Size: 30000     Sorted: True      Time Taken = 16001

Process returned 0 (0x0)   execution time : 0.113 s
Press any key to continue.
```

Average Case:



Best Case:



Worst Case: