Insertion Sort ()

#include <bits/stdc++.h>  
using namespace std;  
void insertion\_sort(int arr[], int n) {  
 for (int i = 1; i < n; i++) {  
 int k = arr[i];  
 int j = i - 1;  
 while (j >= 0 && arr[j] > k) {  
 arr[j + 1] = arr[j];  
 j = j - 1;  
 }  
 arr[j + 1] = k;  
 }  
}  
int main() {  
 const int n = 100'000;  
 int\* arr = new int[n];  
 random\_device rd;  
 mt19937 mt(rd());  
 uniform\_int\_distribution<int> dis(0, 999'999);  
 for (int i = 0; i < n; i++) {  
 arr[i] = dis(mt);  
 }  
 insertion\_sort(arr, n);  
 return 0;  
}

На компјутер со спецификации: Intel Core i5-9400F @2.90GHz, Ram Memory 8GB, SSD 256GB, HDD 1TB, NVIDIA GeForce GTX 1050 Ti

Програмата се изврши за 160ms, 75ms, 56ms, 82ms, 55ms

Selection Sort ()

#include <bits/stdc++.h>  
using namespace std;  
void selectionSort(int arr[], int n) {  
 for (int i = 0; i < n-1; i++) {  
 int min = i;  
 for (int j = i+1; j < n; j++) {  
 if (arr[min] > arr[j]) {  
 min = j;  
 }  
 }  
 swap(arr[i], arr[min]);  
 }  
}  
int main() {  
 const int n = 100'000;  
 int\* arr = new int[n];  
 random\_device rd;  
 mt19937 mt(rd());  
 uniform\_int\_distribution<int> dis(0, 999'999);  
 for (int i = 0; i < n; i++) {  
 arr[i] = dis(mt);  
 }  
 selectionSort(arr, n);  
 return 0;  
}

На компјутер со спецификации: Intel Core i5-9400F @2.90GHz, Ram Memory 8GB, SSD 256GB, HDD 1TB, NVIDIA GeForce GTX 1050 Ti

Програмата се изврши за 105ms, 147ms, 65ms, 219ms, 54ms

Quick Sort ( или O(n) во најлош случај, O(n log n) во најдобар случај)

#include <bits/stdc++.h>  
using namespace std;  
int pat(int arr[],int low, int high){  
 int piv = arr[high];  
 int i = (low-1);  
 for (int j = low; j <= high-1; j++) {  
 if (arr[j] <= piv) {  
 i++;  
 swap(arr[i], arr[j]);  
 }  
 }  
 swap(arr[i+1], arr[high]);  
 return (i+1);  
}  
void quickSort(int arr[],int low, int high) {  
 if (low < high) {  
 int pi = pat(arr, low, high);  
 quickSort(arr, low, pi-1);  
 quickSort(arr, pi+1, high);  
 }  
}  
int main() {  
 const int n = 100'000;  
 int\* arr = new int[n];  
 random\_device rd;  
 mt19937 mt(rd());  
 uniform\_int\_distribution<int> dis(0, 999'999);  
 for (int i = 0; i < n; i++) {  
 arr[i] = dis(mt);  
 }  
 quickSort(arr, 0, n-1);  
 return 0;  
}

На компјутер со спецификации: Intel Core i5-9400F @2.90GHz, Ram Memory 8GB, SSD 256GB, HDD 1TB, NVIDIA GeForce GTX 1050 Ti

Програмата се изврши за 163ms, 61ms, 54ms, 59ms, 53ms

Merge Sort (O(n log n))

#include <bits/stdc++.h>  
using namespace std;  
void mer(int arr[],int l, int m, int r) {  
 int n1 = m - l + 1;  
 int n2 = r - m;  
 int\* L = new int[n1];  
 int\* R = new int[n2];  
 for (int i = 0; i < n1; i++) {  
 L[i] = arr[l + i];  
 }  
 for (int i = 0; i < n2; i++) {  
 R[i] = arr[m + 1 + i];  
 }  
 int i = 0, j = 0, k = 1;  
 while (i < n1 && j < n2) {  
 if (L[i] <= R[j]) {  
 arr[k++] = L[i++];  
 }  
 else {  
 arr[k++] = R[j++];  
 }  
 }  
 while (i < n1) {  
 arr[k++] = L[i++];  
 }  
 while (j < n2) {  
 arr[k++] = R[j++];  
 }  
 delete[] L;  
 delete[] R;  
}  
void mergesort(int arr[], int l, int r) {  
 if (l<r) {  
 int m = l+(r-1)/2;  
 mergesort(arr, l, m);  
 mer(arr,l,m,r);  
 }  
}  
int main() {  
 const int n = 100'000;  
 int\* arr = new int[n];  
 random\_device rd;  
 mt19937 mt(rd());  
 uniform\_int\_distribution<int> dis(0, 999'999);  
 for (int i = 0; i < n; i++) {  
 arr[i] = dis(mt);  
 }  
 mergesort(arr, 0, n-1);  
 return 0;  
}

На компјутер со спецификации: Intel Core i5-9400F @2.90GHz, Ram Memory 8GB, SSD 256GB, HDD 1TB, NVIDIA GeForce GTX 1050 Ti

Програмата се изврши за 80ms, 54ms, 58ms, 58ms, 56ms