Codes

Processor: Intel i7-4510U 2.00 GHz, 64bit

RAM: 8GB

Runs: За n=1000;

Quicksort:

#include <bits/stdc++.h>

using namespace std;

void qs(int a[], int dolzina);

void quicksort(int a[], int n, int m);

int p(int a[], int n, int m);

int main() {

const int MAXN = 10000;

int x[MAXN];

int dolzina;

cin >> dolzina;

for (int i = 0; i < dolzina; i++) {

cin >> x[i];

}

qs(x, dolzina);

for (int i = 0; i < dolzina; i++) {

cout << x[i] << " ";

}

return 0;

}

void qs(int a[], int dolzina) {

quicksort(a, 0, dolzina - 1);

}

void quicksort(int a[], int n, int m) {

if (n < m) {

int p\_index = p(a, n, m);

quicksort(a, n, p\_index - 1);

quicksort(a, p\_index + 1, m);

}

}

int p(int a[], int n, int m) {

int p\_value = a[m];

int i = n;

for (int j = i; j < m; j++) {

if (a[j] <= p\_value) {

swap(a[i], a[j]);

i++;

}

}

swap(a[i], a[m]);

return i;

}

Јас кога го ранав кодот му требаше:

Run 1: 0.192ms

Run 2: 0.128ms

Run 3: 0.063ms

MergeSort:

#include <bits/stdc++.h>

using namespace std;

void merge(int A[], int low, int mid, int high) {

int i = low, j = mid + 1, k = low;

static int c[10000];

while (i <= mid && j <= high) {

if (A[i] < A[j]) {

c[k++] = A[i++];

} else {

c[k++] = A[j++];

}

}

while (i <= mid) {

c[k++] = A[i++];

}

while (j <= high) {

c[k++] = A[j++];

}

for (int p = low; p < k; p++) {

A[p] = c[p];

}

}

void mergesort(int A[], int low, int high) {

if (low < high) {

int mid = (low + high) / 2;

mergesort(A, low, mid);

mergesort(A, mid + 1, high);

merge(A, low, mid, high);

}

}

int main() {

const int MAXN = 10000;

int A[MAXN];

int n;

cin >> n;

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

cin >> A[i];

}

mergesort(A, 0, n - 1);

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

cout << A[i] << " ";

}

cout << endl;

return 0;

}

Run 1: 0.271ms

Run 2: 0.216ms

Run 3: 0.173ms

InsertionSort

#include <bits/stdc++.h>

using namespace std;

void is(int a[], int dolzina) {

for (int i = 1; i < dolzina; i++) {

int k = a[i];

int j = i - 1;

while (j >= 0 && a[j] > k) {

a[j + 1] = a[j];

j = j - 1;

}

a[j + 1] = k;

}

}

int main() {

const int MAXN = 10000;

int a[MAXN];

int dolzina;

cin >> dolzina;

for (int i = 0; i < dolzina; i++) {

cin >> a[i];

}

is(a, dolzina);

for (int i = 0; i < dolzina; i++) {

cout << a[i] << " ";

}

cout << endl;

return 0;

}

Run 1: 2.33ms

Run 2: 1.97ms

Run 3: 1.43ms

SelectionSort

#include <bits/stdc++.h>

using namespace std;

void ss(int a[], int dolzina) {

for (int i = 0; i < (dolzina - 1); i++) {

int min\_pos = i;

for (int j = i + 1; j < dolzina; j++) {

if (a[j] < a[min\_pos]) {

min\_pos = j;

}

}

if (min\_pos != i) {

int temp = a[i];

a[i] = a[min\_pos];

a[min\_pos] = temp;

}

}

}

int main() {

const int MAXN = 10000;

int a[MAXN];

int n;

cin >> n;

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

cin >> a[i];

}

ss(a, n);

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

cout << a[i] << " ";

}

cout << endl;

return 0;

}

Run 1: 2.53ms

Run 2: 2.01ms

Run 3: 1.93ms