

Δομές Δεδομένων και Αλγόριθμοι (Γ' εξάμηνο)

Τμήμα Μηχανικών Πληροφορικής ΤΕ, ΤΕΙ Ηπείρου

Γκόγκος Χρήστος

Άσκηση εργαστηρίου #1 (ορεκτικό)

Ο ακόλουθος κώδικας δημιουργεί ένα πρόγραμμα που δίνει 2 επιλογές στον χρήστη.

1. Generate data (δημιουργία δεδομένων)
 2. Read and sort data (ανάγνωση και ταξινόμηση δεδομένων)
- What do you want to do?

Αν επιλέξει την επιλογή 1 τότε δημιουργείται ένα αρχείο με όνομα data.txt που θα περιέχει 1000 τυχαίες ακέραιες τιμές στο διάστημα [0, 5000). Αν ο χρήστης επιλέξει την επιλογή 2 τότε διαβάζεται το αρχείο από τον σκληρό δίσκο τοποθετείται σε ένα πίνακα ακεραίων και εν συνεχεία ταξινομείται με την μέθοδο ταξινόμησης με εισαγωγή (insertion_sort).

```
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <ctime>
using namespace std;

void generate_data(int n);
void read_sort_data();
void insertion_sort(int a[], int n);
void print_array(int a[], int n);

int main(int argc, char **argv) {
    int choice;
    cout << "1. Generate data" << endl;
    cout << "2. Read and sort data" << endl;
    cout << "What do you want to do? ";
    cin >> choice;
    if (choice == 1)
        generate_data(1000);
    else if (choice == 2)
        read_sort_data();
}

void generate_data(int n) {
    ofstream filestr("data.txt");
    if (filestr.is_open()) {
        srand(time(0));
        for (int i = 0; i < n; i++) {
            int r = rand() % 5000;
            filestr << r << "\n";
        }
    }
}
```

```

        }
        filestr.close();
        cout << "generated " << n << " values"<< endl;
    } else {
        cerr << "error" << endl;
    }
}

void read_sort_data() {
    string buffer;
    int c = 0;
    fstream filestr("data.txt");
    if (filestr.is_open()) {
        while (getline(filestr, buffer))
            c++;
        filestr.close();

        int a[c];
        filestr.open("data.txt");
        int i = 0;
        while (getline(filestr, buffer)) {
            a[i] = atoi(buffer.c_str());
            i++;
        }
        filestr.close();
        insertion_sort(a, c);
        print_array(a, c);
    } else {
        cerr << "error" << endl;
    }
}

void insertion_sort(int a[], int n) {
    for (int i = 1; i < n; i++) {
        int key = a[i];
        int j = i - 1;
        while ((j >= 0) && (key < a[j])) {
            a[j + 1] = a[j];
            j--;
        }
        a[j + 1] = key;
    }
}

void print_array(int a[], int n) {
    for (int i=0;i<n;i++)
        printf("%d --> %d\n", i, a[i]);
}

```

Ο παραπάνω κώδικας μπορεί να αντιγραφεί από την ακόλουθη διεύθυνση:

http://chggogos.github.io/ce_teiep_dsa2014_exercise01.html

Ερώτημα 1

Προσθέστε μια επιπλέον συνάρτηση με όνομα `selection_sort` που να υλοποιεί τον αλγόριθμο ταξινόμησης με επιλογή. Τροποποιήστε τον κώδικα έτσι ώστε να καλεί την νέα συνάρτηση ταξινόμησης. Επιβεβαιώστε ότι τα αποτελέσματα της ταξινόμησης είναι σωστά.

Ερώτημα 2

Κατασκευάστε πίνακα συχνοτήτων που να υπολογίζει πόσες φορές υπάρχει η κάθε μια τιμή μέσα στο αρχείο και εμφανίστε τον στην οθόνη. Για να γίνει αυτό θα χρειαστεί να προσθέσετε μια νέα επιλογή στο αρχικό μενού.

Ερώτημα 3

Χρονομετρήστε τον χρόνο που χρειάζεται για να ταξινομήσει τα δεδομένα η ταξινόμηση με επιλογή και τον χρόνο που χρειάζεται η ταξινόμηση με εισαγωγή. Λόγω του ότι η ταξινόμηση 1000 μόνο τιμών αναμένεται να είναι πολύ γρήγορη τροποποιήστε τον κώδικα έτσι ώστε να πραγματοποιεί ταξινόμηση σε 50000 τιμές (εφόσον πρώτα τις δημιουργήσετε). Για να μετρήσετε τον χρόνο που απαιτεί η ταξινόμηση μπορείτε να συμβουλευτείτε τον ακόλουθο κώδικα ο οποίος μετράει τον χρόνο που απαιτεί ο υπολογισμός των τετραγωνικών ριζών όλων των ακεραίων αριθμών από το 1 μέχρι το 10^6 .

```
#include <ctime>
...
clock_t t1, t2;
t1 = clock();

double sum = 0.0;
for (int i=1; i<=1000000; i++)
    sum += sqrt(i);
cout << sum << endl;

t2 = clock();
double elapsed_time = (double) (t2-t1)/CLOCKS_PER_SEC;
cout << "Elapsed time " << elapsed_time << endl;
```

Ερώτημα 4

Χρησιμοποιήστε τη συνάρτηση ταξινόμησης `sort` που εμπεριέχεται στη C++. Ποιος είναι ο χρόνος που απαιτείται τώρα για την ταξινόμηση των δεδομένων; Συμβουλευτείτε τον ακόλουθο κώδικα ο οποίος ταξινομεί έναν πίνακα 5 θέσεων για το πώς λειτουργεί η εν λόγω συνάρτηση.

```
#include <iostream>
#include <algorithm>
using namespace std;

int main(int argc, char **argv) {
    cout << "### STL Sort Example ###" << endl;
    int a[] = { 17, 11, 8, 16, 15 };
}
```

```
cout << "BEFORE: ";  
for (int i = 0; i < 5; i++)  
    cout << a[i] << " ";  
cout << endl;  
sort(a, a + 5);  
cout << "AFTER: ";  
for (int i = 0; i < 5; i++)  
    cout << a[i] << " ";  
cout << endl;  
return 0;  
}
```