

# PROJECT-ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ 2020

|                             |            |
|-----------------------------|------------|
| Καπετανίδης Παναγιώτης      | AM:1067426 |
| Λιούμη Κυριακή              | AM:1067410 |
| Κυριάκος Βασίλειος-Γεώργιος | AM:1070915 |
| Καλίωρας Φώτιος             | AM:1067408 |

## ΕΙΣΑΓΩΓΗ

Ο στόχος αυτού του pdf είναι να αναλυθεί και να γίνει κατανοητός ο τρόπος συγγραφής καθώς και η λειτουργία των προγραμμάτων. Παρακάτω ακολουθούν τα εξής 3 μέρη:

1. Φιλοσοφία κώδικα.
2. Απαντήσεις ερωτημάτων θεωρίας.
3. Ανάλυση της λειτουργίας του τελικού μενού στο δεύτερο μέρος.

Η συγγραφή του προγράμματος έγινε σε **C++**, η γλώσσα αυτή μας επιτρέπει να αξιοποιήσουμε εργαλεία όπως ο αντικειμενοστραφής προγραμματισμός και οι δυναμικές δομές με σχετική ευκολία.

## Παράρτημα 1

Τα προγράμματα χωρίζονται σε 3 τμήματα:

1. *Τη διαχείριση των αρχείων και πληροφορίας.*
2. *Τον ζητούμενο αλγόριθμο.*
3. *Την συνάρτηση main.*

Κάθε πρόγραμμα που έχουμε φτιάξει βασίζεται στην ίδια φιλοσοφία (με μικρές τροποποιήσεις τους σύμφωνα με τις ανάγκες τόσο του ερωτήματος, όσο και του ζητούμενου αλγορίθμου).

Προσπαθήσαμε να δημιουργήσουμε κώδικα ο οποίος είναι όσο το δυνατόν πιο επεκτάσιμος, για αυτόν τον λόγο ακριβώς πολλά κομμάτια κώδικα επαναλαμβάνονται μέσα στα διάφορα προγράμματα.

## Ανάλυση τμημάτων

### 1. Διαχείριση αρχείων και πληροφορίας

Βασικό πρόβλημα τίθεται όταν πρέπει να διαβάσουμε και να επεξεργαστούμε πληροφορίες οι οποίες προέρχονται από αρχεία. Στην εργασία αυτή καλούμαστε να διαβάσουμε αυτές τις πληροφορίες, αλλά και να τις ομαδοποιήσουμε.

Για να πετύχουμε το σκοπό αυτό, δημιουργήσαμε δύο συναρτήσεις: η πρώτη μετατρέπει τις πληροφορίες του αρχείου σε string και η δεύτερη διαχειρίζεται αυτά τα strings, τα μετασχηματίζει στη ζητούμενη μορφή (π.χ int,long,string) και τέλος τα αποθηκεύει σε ένα πίνακα από δομές τύπου temp(θερμοκρασία),hum(υγρασία),day(ημέρα).

Η κάθε δομή αποτελείται από τουλάχιστον δύο μεταβλητές, την ημερομηνία καθώς και την τιμή της θερμοκρασίας/υγρασίας.

Ορισμένες δομές απαιτούν παραπάνω μεταβλητές εφόσον το έχει ανάγκη το πρόγραμμα.

Η επιλογή της κάθε δομής εξαρτάται από το ερώτημα καθώς και το αρχείο το οποίο διαχειριζόμαστε.

### 2. Ο ζητούμενος αλγόριθμος

Σε κάθε πρόγραμμα ο ζητούμενος αλγόριθμος είναι διαφορετικός, ωστόσο ο τρόπος με τον οποίο αντιμετωπίσαμε τους αλγορίθμους είναι παρόμοιος.

Σε κάθε αλγόριθμο περνάμε σαν όρισμα τον πίνακα δομών ο οποίος περιέχει τη πληροφορία.

Στόχος μας είναι η επεξεργασία των δεδομένων μέσα από τις μεταβλητές της εκάστοτε δομής(δηλαδή κυψελίδας του πίνακα δομών) και η επιστροφή των αποτελεσμάτων, ξανά σε μορφή δομής.

### 3. Η συνάρτηση main

Στη συνάρτηση main καλούμε τις διάφορες συναρτήσεις οι οποίες εξυπηρετούν τη λειτουργία του προγράμματος αφού πρώτα έχουμε δημιουργήσει και αρχικοποιήσει τις απαιτούμενες μεταβλητές και δομές.

## Παράρτημα 2

- **Ερώτημα 1:**

Οι αλγόριθμοι **Merge Sort** και **Quick Sort** έχουν παρόμοιο χρόνο εκτέλεσης στη μέση περίπτωση  **$O(n \log n)$** .

Mergesort = **1.4sec** και Quicksort = **1.34sec**.

- **Ερώτημα 2:**

Οι αλγόριθμοι **Heap Sort** και **Counting Sort** έχουν παρόμοιο χρόνο εκτέλεσης  $t =$  **0.84sec**.

- **Ερώτημα 3:**

Για τους αλγορίθμους **Δυαδικής Αναζήτησης** και **Αναζήτησης με Παρεμβολή** έχουμε παρόμοιο χρόνο εκτέλεσης (Μετράμε μόνο την αναζήτηση)

Δυαδική Αναζήτηση =  **$8e-05s$**

Αναζήτηση και Παρεμβολή =  **$7.7e-05s$**

Για μη ισοπίθανες κατανομές επηρεάζεται δραματικά η απόδοση του αλγορίθμου πχ. **Αλλάζοντας την τελευταία τιμή** του hum.txt σε 10.000 έχει σαν αποτέλεσμα **την αλλαγή τάξης του χρόνου** εκτέλεσης κατά  **$10^2(0.000128)$** .

- **Ερώτημα 4:**

**Μέση/χειρότερη** =  $\log(\log n)/\sqrt{n} = \log(\log(x)/\log(2))/(\sqrt{x} \log(2))$ .

Για  $x=8222$  (αριθμός στοιχείων) **αναμένουμε αποτέλεσμα της τάξης του 0.1**, πράγματι για την ακραία ημερομηνία 9999999 με χρόνο 0.0001 και τη μέση περίπτωση  $7.9e-05$  έχουμε λόγο=0.6, ο οποίος **είναι της τάξης του 0.1**.

Για τον κανονικό αλγόριθμο του BIS έχουμε  **$O(n^{1/2})$**  στη χειρότερη περίπτωση ενώ **για τη παραλλαγή του** έχουμε  **$O(\log n)$**  στη χειρότερη περίπτωση.

**BIS/παραλλαγή BIS=  $O(n^{1/2}) / O(\log n)$ .**

Πράγματι η παραλλαγή BIS είναι **10 φορές πιο γρήγορη** και στην πράξη για τη χειρότερη περίπτωση.

## Παράρτημα 3

### Ανάλυση της λειτουργίας του τελικού μενού

Στο δεύτερο μέρος μας ζητείται να υλοποιήσουμε ένα πλήθος εφαρμογών, των οποίων η κλήση πραγματοποιείται σε ένα τελικό ενιαίο μενού.

Η τελική εφαρμογή έχει τη δυνατότητα διαχείρισης πληροφορίας μέσω **Hash Chain** καθώς και **Binary Search Trees**.

Και οι δύο μέθοδοι αντιστοιχούν σε μια κλάση και το πρόγραμμα λειτουργεί με βάση τα αντικείμενα τα οποία προκύπτουν από τις παραπάνω κλάσεις.

Φυσικά η *main* και πάλι διασφαλίζει την επικοινωνία κλάσεων, συναρτήσεων και δομών.

Στον χρήστη δίνονται οι επιλογές αξιοποίησης αυτών των κλάσεων και τον αντίστοιχων μενού για την ανάκτηση και τροποποίηση της πληροφορίας.