PROJECT: "DATA STRUCTURES 2023"

PART I: "Sorting and Searching Algorithms"

Στην ιστοσελίδα **https://www.stats.govt.nz/large-datasets/csv-files-for-download/** υπάρχουν δεδομένα για ανάλυση/επεξεργασία σε αρχεία csv που αφορούν τους ακόλουθους τομείς: Business, Census, Economy, Effects of COVID-19 on trade , Environment, Government finance, Health, Industries, Labour market, Population, Society. Στην παρούσα εργασία θα ασχοληθούμε με δεδομένα από το Effects of COVID-19 on trade, και συγκεκριμένα το "dataset effects-of-covid-19-on-trade-at-15-december-2021-provisional.csv[1]", το οποίο περιέχει συνολικά 111.438 εγγραφές, με την ακόλουθη δομή:

Direction Year Date Weekday Country Commodity Transport_Mode Measure Value Cumulative

Το παραπάνω αρχείο λοιπόν περιέχει τα ακόλουθα πεδία: Direction {imports, exports , reimports}, Year {2015,…., 2021}, Date {01/01/2015, 01/02/2015,……, 15/12/2021}, Weekday {Monday,…., Sunday}, Country {All, China, European Union, Asia, Australia, USA,…..}, Commodity {"All",
"Milk powder, butter, and cheese", "Fish, crustaceans, and molluscs", "Non-food manufactured goods", "Electrical machinery and equip",…….}, Transport_Mode {All, sea, air, …}, Measure { $, Tonnes, …}, Value {long integer της τάξης των εκατομυρίων}, Cumulative {long integer της τάξης των εκατομυρίων...}.

Σας ζητείται να υλοποιήσετε τέσσερα διαφορετικά προγράμματα σε γλώσσα της επιλογής σας (κατά προτίμηση C, C++), που να χρησιμοποιούν ως είσοδο το παραπάνω αρχείο και το καθένα να υλοποιεί τις παρακάτω λειτουργίες:

(1) Ταξινόμηση κατά αύξουσα σειρά των ημερομηνιών (Πεδίο **Date**) βάσει των τιμών του Πεδίου **Value** κάνοντας χρήση των αλγορίθμων **Counting Sort** και **merge Sort**, σύμφωνα με τον ψευδοκώδικα που σας επεξηγήθηκε στη θεωρία (για λεπτομέρειες δείτε τις σχετικές διαφάνειες στο e-class). Συγκρίνατε πειραματικά τους δύο (2) αλγορίθμους. Τι παρατηρείτε?

(2) Ταξινόμηση κατά αύξουσα σειρά των ημερομηνιών βάσει των τιμών **Cumulative** κάνοντας χρήση των αλγορίθμων **Heap Sort** και **Quick Sort**, σύμφωνα με τον ψευδοκώδικα που σας επεξηγήθηκε στη θεωρία (για λεπτομέρειες δείτε τις σχετικές διαφάνειες στο e-class). Συγκρίνατε πειραματικά τους δύο (2) αλγορίθμους. Τι παρατηρείτε?

(3) Εύρεση **value** ή/και **cumulative** για συγκεκριμένη ημερομηνία (**Date**) που θα δίνεται από το χρήστη, σύμφωνα με τους αλγορίθμους **Δυαδικής Αναζήτησης** και **Αναζήτησης με Παρεμβολή.** Τί παρατηρείτε ως προς τους χρόνους μέσης περίπτωσης? Πόσο η ΚΑΤΑΝΟΜΗ του Data Set επηρεάζει την απόδοση του κάθε αλγορίθμου?

(4) Υλοποιήστε το ζητούμενο του ερωτήματος (3) κάνοντας χρήση του αλγορίθμου **Δυικής Αναζήτησης Παρεμβολής (BIS)**. Συμβουλευτείτε τον ψευδοκώδικα της σελίδας 80 του βιβλίου «Δομές Δεδομένων», Α.Κ. Τσακαλίδης, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής καθώς και τις διαφάνειες *searching.pdf* που είναι διαθέσιμα στο e-class. Επαληθεύστε πειραματικά τη χρονική πολυπλοκότητα που ισχύει για την μέση (expected) και χειρότερη περίπτωση (worst-case).Η βελτίωση της χειρότερης περίπτωσης επιτυγχάνεται με μία παραλλαγή του BIS. Συμβουλευτείτε τη σελίδα 83 του βιβλίου «Δομές Δεδομένων», Α.Κ. Τσακαλίδης, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής καθώς και τις

[1] Source: Stats NZ and licensed by Stats NZ for reuse under the Creative Commons Attribution 4.0 International licence.

διαφάνειες *searching.pdf* που είναι διαθέσιμα στο e-class και υλοποιήστε τον αλγόριθμο της συγκεκριμένης παραλλαγής του **BIS.** Συγκρίνατε πειραματικά τους παραπάνω δύο αλγορίθμους. Τί παρατηρείτε ως προς τους χρόνους χειρότερης περίπτωσης?

PART II: "BSTs && HASHING"

Με τον κατάλληλο ορισμό δομών (structs) και συναρτήσεων (functions), να υλοποιήσετε μια εφαρμογή (να γράψετε ένα πρόγραμμα σε γλώσσα C) που θα επεξεργάζεται τα δεδομένα του αρχείου **"**dataset effects-of-covid-19-on-trade-at-15-december-2021-provisional**.csv"**. Θυμίζουμε ξανά ότι κάθε γραμμή του αρχείου αυτού αντιστοιχεί σε μία ημέρα μετρήσεων, ενώ οι γραμμές έχουν την παρακάτω μορφή:

Direction Year Date Weekday Country Commodity Transport_Mode Measure Value Cumulative

(A) Η εφαρμογή διαβάζει αρχικά το αρχείο και δημιουργεί ένα **Δένδρο BST** στο οποίο κάθε κόμβος του διατηρεί την εγγραφή (Date, Value της ημέρας αυτής). Το δέντρο που θα υλοποιήσετε θα είναι ισοζυγισμένο της δικής σας επιλογής (AVL δέντρο, (2,4) δέντρο ή γενικά (a,b)-tree με τα a και b οριζόμενα από εσάς). Το **BST** διατάσσεται ως προς την ΗΜΕΡΟΜΗΝΙΑ (Date) και υλοποιείται με δυναμική διαχείριση μνήμης. Μετά την δημιουργία του **BST** η εφαρμογή εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. **Απεικόνιση** του **BST** με ενδο-διατεταγμένη διάσχιση. Κάθε απεικόνιση θα πρέπει να περιέχει μια επικεφαλίδα με τους τίτλους των στοιχείων των εγγραφών που απεικονίζονται.
2. **Αναζήτηση** της τιμής Value βάσει ΗΜΕΡΟΜΗΝΙΑΣ (Date) που θα δίνεται από το χρήστη.
3. **Τροποποίηση** του περιεχομένου του πεδίου value που αντιστοιχεί σε συγκεκριμένη ΗΜΕΡΟΜΗΝΙΑ (Date).
4. **Διαγραφή** μιας εγγραφής που αντιστοιχεί σε συγκεκριμένη ΗΜΕΡΟΜΗΝΙΑ (Date).
5. **Έξοδος** από την εφαρμογή.

(Β) Τροποποιήστε κατάλληλα τον κώδικα του (Α), ώστε το αρχείο να διαβάζεται στο **BST** με βάση την τιμή του πεδίου value. Το **BST** διατάσσεται ως προς την τιμή Value και υλοποιείται με δυναμική διαχείριση μνήμης. Μετά την δημιουργία του **BST** η εφαρμογή εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. Εύρεση ΗΜΕΡΑΣ/ΗΜΕΡΩΝ με την ΕΛΑΧΙΣΤΗ ΤΙΜΗ Value.
2. Εύρεση ΗΜΕΡΑΣ/ΗΜΕΡΩΝ με τη ΜΕΓΙΣΤΗ ΤΙΜΗ Value.

(Γ) Υλοποιήστε το (Α) κάνοντας χρήση HASHING με αλυσίδες, αντί **BST**. Η συνάρτηση κατακερματισμού θα υπολογίζεται ως το υπόλοιπο (modulo) της διαίρεσης του αθροίσματος των κωδικών ASCII των επιμέρους χαρακτήρων που απαρτίζουν την ΗΜΕΡΟΜΗΝΙΑ με ένα περιττό αριθμό m που συμβολίζει το πλήθος των κάδων (buckets). Π.χ. για ΗΜΕΡΟΜΗΝΙΑ=" 15/12/2021" και m=11, ισχύει:

Hash("15/12/2021")= [ASCII('1')+ ASCII('5')+ ASCII('/')+ ASCII('1')+ ASCII('2')+ ASCII('/')+ ASCII('2')+ ASCII('0')+ ASCII('2')+ ASCII('1')] mod 11.

Το πρόγραμμα θα εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. **Αναζήτηση** Τιμής Value βάσει της ΗΜΕΡΟΜΗΝΙΑΣ που θα δίνεται από το χρήστη.
2. **Τροποποίηση** των στοιχείων εγγραφής βάσει ΗΜΕΡΟΜΗΝΙΑΣ που θα δίνεται από το χρήστη. Η τροποποίηση προφανώς αφορά ΜΟΝΟ το πεδίο Value.

3. **Διαγραφή** μιας εγγραφής από τον πίνακα κατακερματισμού βάσει ΗΜΕΡΟΜΗΝΙΑΣ που θα δίνεται από το χρήστη.
4. **Έξοδος** από την εφαρμογή.

Ενοποιήστε τα (Α), (Β) και (Γ) σε ένα πρόγραμμα στο οποίο ο χρήστης θα ερωτάται αν θέλει τη φόρτωση του αρχείου σε ένα **BST** ή σε μία δομή **Hashing** με αλυσίδες και στην περίπτωση που ο χρήστης επιλέξει το πρώτο να μπορεί εν συνεχεία να επιλέξει αν η φόρτωση στο **BST** θα γίνει με βάση την ΗΜΕΡΟΜΗΝΙΑ ή την τιμή Value.

DEADLINE: ΤΡΕΙΣ ΜΕΡΕΣ ΠΡΙΝ ΤΗΝ ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ ΕΑΡΙΝΟΥ ΕΞΑΜΗΝΟΥ

Η παράδοση της άσκησης θα πραγματοποιείται με ΑΝΑΡΤΗΣΗ ΣΤΟ E_CLASS και με αποστολή μηνύματος ηλεκτρονικού ταχυδρομείου ΚΑΙ ΣΤΙΣ ΤΡΕΙΣ ακόλουθες διευθύνσεις με ένα μήνυμα (με τρεις παραλήπτες και όχι τρια διακριτά μηνύματα): **sioutas@ceid.upatras.gr, makri@ceid.upatras.gr, mvonitsanos@ceid.upatras.gr,**

Μπορείτε να συντάξετε την αναφορά σας σε όποια μορφή κειμένου επιθυμείτε (word, pdf, κ.λπ.). Στο ηλεκτρονικό μήνυμα που θα αποστείλετε θα έχετε συμπιεσμένα το αρχείο της αναφοράς σας καθώς και τα αρχεία των προγραμμάτων C.

ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ ΑΝΑ ΟΜΑΔΑ <=4

ΠΟΣΟΣΟΣΤΟ ΕΠΙ ΤΟΥ ΣΥΝΟΛΙΚΟΥ ΒΑΘΜΟΥ: 30%

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!!