PROJECT: "DATA STRUCTURES 2024"

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 **Population**, και συγκεκριμένα το "**bd-dec22-births-deaths-by-region.csv**", το οποίο περιέχει εγγραφές, με την ακόλουθη δομή:

Period	Birth_Death	Region	Count

Που αφορούν τον αριθμό γεννήσεων/θανάτων από το 2005 έως και το 2022 ανά περιοχή.

Period IN [2005, 2022], Birth_Death IN {Births, Deaths}, Region: STRING, Count: Integer

Μετατρέψτε το αρχείο csv σε txt και διαβάστε το σε Array of Structs. Εκτελέστε ταξινόμηση με βάση συγκεκριμένο πεδίο του struct που ζητείται στην εκφώνηση και εμφανίστε/τυπώστε το πεδίο του struct που ζητείται από το ταξηνομημένο πλέον array και που μπορεί να είναι διαφορετικό από το πεδίο στο οποίο βασίστηκε η ταξινόμηση.

Σας ζητείται να υλοποιήσετε τέσσερα διαφορετικά προγράμματα σε γλώσσα C/C++/Java που να χρησιμοποιούν ως είσοδο το παραπάνω αρχείο και το καθένα να υλοποιεί τις παρακάτω λειτουργίες:

(1) Ταξινόμηση κατά αύξουσα σειρά των περιοχών (Region) με βάση τις τιμές των **αριθμών γεννήσεων** (Count) κάνοντας χρήση των αλγορίθμων **Merge 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?

(2) Ταξινόμηση κατά αύξουσα σειρά των των περιοχών (Region) με βάση τις τιμές των **αριθμών θανάτων** κάνοντας χρήση των αλγορίθμων **Heap Sort** και **Counting 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 περιοχών (Region) με βάση τον **αριθμό γεννήσεων** ο οποίος θα ανήκει σε εύρος τιμών **[b1, b2],** 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? [**ΣΗΜΕΙΩΣΗ**: Αναζητείστε πρώτα το αριστερό άκρο του εύρους τιμών, b1, και σαρώστε σειριακά το ταξηνομημένο array of structs μέχρι να συναντήσετε το δεξιό άκρο b2]

(4) Υλοποιήστε το ζητούμενο του ερωτήματος (3) κάνοντας χρήση του αλγορίθμου **Δυικής Αναζήτησης Παρεμβολής (BIS)**. Συμβουλευτείτε τον ψευδοκώδικα της σελίδας 82 του βιβλίου «Δομές Δεδομένων», Α.Κ. Τσακαλίδης, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής καθώς και τις διαφάνειες *9.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 85 του βιβλίου «Δομές Δεδομένων», Α.Κ. Τσακαλίδης, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής καθώς και τις διαφάνειες *9.searching.pdf* που είναι διαθέσιμα στο e-class και υλοποιείστε 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/C++/Java) που θα επεξεργάζεται τα δεδομένα του αρχείου bd-dec22-births-deaths-by-region.csv. Θυμίζουμε ξανά ότι κάθε γραμμή του αρχείου αυτού αντιστοιχεί σε ημερολογιακά έτη μετρήσεων γεννήσεων/θανάτων από το 2005 έως το 2022, ενώ οι γραμμές έχουν την παρακάτω μορφή:

{Period, Birth_Death, Region, Count}

(Α) Η εφαρμογή διαβάζει αρχικά το αρχείο και δημιουργεί ένα **Δυαδικό Δένδρο Αναζήτησης** (ΔΔΑ) στο οποίο κάθε κόμβος του διατηρεί την εγγραφή **(Region, Period, Count_of_Births)**. Το ΔΔΑ διατάσσεται ως προς το πεδίο **Region** και υλοποιείται με δυναμική διαχείριση μνήμης. Μετά την δημιουργία του ΔΔΑ η εφαρμογή εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. Απεικόνιση του ΔΔΑ με ενδο-διατεταγμένη διάσχιση. Κάθε απεικόνιση θα πρέπει να περιέχει μια επικεφαλίδα με τον τίτλο της ΠΕΡΙΟΧΗΣ της εγγραφής που απεικονίζεται.
2. Αναζήτηση του αριθμού γεννήσεων για συγκεκριμένη **χρονική περίοδο** και **περιοχή** που θα δίνονται από το χρήστη.
3. Τροποποίηση του πεδίου αριθμού γεννήσεων για συγκεκριμένη **χρονική περίοδο** και **περιοχή** που θα δίνονται από το χρήστη.
4. Διαγραφή μιας εγγραφής βάσει **της περιοχής** που θα δίνεται από το χρήστη.
5. Έξοδος από την εφαρμογή.

(Β) Τροποποιήστε κατάλληλα τον κώδικα του (Α), ώστε το αρχείο να διαβάζεται στο ΔΔΑ με βάση *τον αριθμό γεννήσεων* **(Count_of_Births, Period, Region)**. Το ΔΔΑ διατάσσεται ως προς το πεδίο Count_of_Births και υλοποιείται με δυναμική διαχείριση μνήμης. Μετά την δημιουργία του ΔΔΑ η εφαρμογή εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. Εύρεση Περιοχής/Περιοχών με τον ΕΛΑΧΙΣΤΟ ΑΡΙΘΜΟ ΓΕΝΝΗΣΕΩΝ.
2. Εύρεση Περιοχής/Περιοχών με το ΜΕΓΙΣΤΟ ΑΡΙΘΜΟ ΓΕΝΝΗΣΕΩΝ.

(Γ) Υλοποιήστε το (Α) κάνοντας χρήση HASHING με αλυσίδες, αντί ΔΔΑ. Η συνάρτηση κατακερματισμού θα υπολογίζεται ως το υπόλοιπο (modulo) της διαίρεσης του αθροίσματος των κωδικών ASCII των επιμέρους χαρακτήρων που απαρτίζουν την ΠΕΡΙΟΧΗ (πεδίο Region) με ένα περιττό αριθμό m που συμβολίζει το πλήθος των κάδων (buckets). Π.χ. για ΠΕΡΙΟΧΗ=”Northland region” και m=11, ισχύει:

Hash("Northland region")=[ASCII(‘N’)+ ASCII(‘o’)+ ASCII(‘r’)+ ……+ASCII(‘i’)+ASCII(‘o’)+ ASCII(‘n’)] mod 11.

Το πρόγραμμα θα εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. Αναζήτηση του αριθμού γεννήσεων για συγκεκριμένη **χρονική περίοδο** και **περιοχή** που θα δίνονται από το χρήστη.
2. Τροποποίηση του πεδίου αριθμού γεννήσεων για συγκεκριμένη **χρονική περίοδο** και **περιοχή** που θα δίνονται από το χρήστη.
3. Διαγραφή μιας εγγραφής βάσει **της περιοχής** που θα δίνεται από το χρήστη.
4. Έξοδος από την εφαρμογή.

[ΣΗΜΕΙΩΣΗ1: Σκεφτείτε να οργανώσετε αποδοτικά την πληροφορία (λίστα ή πίνακα) που αντιστοιχεί στον κάθε κόμβο του ΔΔΑ καθώς και την αλυσίδα του Hash Table ως προς το πεδίο Period. Τι θα κάνατε? Υλοποιήστε το και κερδίστε **bonus 1/2 ολόκληρο βαθμό]**

[ΣΗΜΕΙΩΣΗ2: Σκεφτείτε το δέντρο αναζήτησης να είναι ζυγισμένο (AVL, ή red-black ή (α,β) δέντρα). Τι θα κάνατε? Υλοποιήστε το και κερδίστε **bonus 1/2 ολόκληρο βαθμό]**

Ενοποιήστε τα (Α), (Β) και (Γ) σε ένα πρόγραμμα στο οποίο ο χρήστης θα ερωτάται αν θέλει τη φόρτωση του αρχείου σε ένα ΔΔΑ 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 ΔΔΑ θα γίνει με βάση την ΠΕΡΙΟΧΗ ή τον ΑΡΙΘΜΟ ΓΕΝΝΗΣΕΩΝ ανά ημέρα.

DEADLINE: 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, aristeid@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/C++/Java.

ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ ΑΝΑ ΟΜΑΔΑ <=4

ΠΟΣΟΣΟΣΤΟ ΕΠΙ ΤΟΥ ΣΥΝΟΛΙΚΟΥ ΒΑΘΜΟΥ: 30% + 10%Bonus

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!!