

Θέματα προετοιμασίας ΚΕΦ. 1, 5, 6, 7, 8, 10 στο μάθημα «ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ»

Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστήμιο Ιωαννίνων, Άρτα

τελευταία ενημέρωση: 30/11/2023

Κεφάλαιο 1

1. Απαντήστε με ΣΩΣΤΟ/ΛΑΘΟΣ
 - a. Ένας αλγόριθμος με πολυπλοκότητα χρόνου $O(n \log n)$ εκτελείται ταχύτερα από ότι ένας αλγόριθμος με πολυπλοκότητα $O(n^2)$ για όλες τις περιπτώσεις εισόδου.
 - b. Ο συμβολισμός του μεγάλου O παρέχει ένα κάτω όριο απόδοσης.
 - c. Ο συμβολισμός του μεγάλου Ω παρέχει ένα κάτω όριο απόδοσης.
 - d. Ο συμβολισμός του μεγάλου Θ παρέχει περισσότερη πληροφορία από το συμβολισμό του μεγάλου O .
 - e. Ο συμβολισμός του μεγάλου O , Ω και Θ έχει να κάνει με το ρυθμό αύξησης του χρόνου εκτέλεσης καθώς τα δεδομένα εισόδου γίνονται πολύ μεγάλα.
 - f. Το πρόβλημα της μέγιστης υποακολουθίας (MAXSUBARRAY) μπορεί να επιλυθεί με αλγόριθμο πολυπλοκότητας $O(n^2)$ καθώς και με αλγόριθμο πολυπλοκότητας $O(n)$.
2. Κατατάξτε τις ακόλουθες συναρτήσεις σε αύξουσα σειρά με βάση το ρυθμό αύξησής τους:
 - a. $\log n$
 - b. n
 - c. n^2
 - d. $n^{1/2}$
 - e. 2^n
 - f. $n!$
 - g. n^3
 - h. $\log \log n$
 - i. 1000
 - j. $n \log n$
3. Ποιες από τις ακόλουθες συναρτήσεις είναι $O(n^2)$;
 - a. n
 - b. $n^{1/2}$
 - c. $2n^2$
 - d. $n^2 + 1000n$
 - e. $n^2/100 + 1$
 - f. $n^2 + n + n \log n$
 - g. n^3
 - h. $n^3 + n^2$
 - i. 2^n
 - j. $n + \log n$
4. Τι χρονική πολυπλοκότητα έχει ο αλγόριθμος εύρεσης της μεγαλύτερης τιμής σε μια ταξινομημένη ακολουθία n θέσεων που επιστρέφει την πρώτη τιμή του πίνακα;
5. Ποια χρονική πολυπλοκότητα χρησιμοποιείται συνήθως;
 - a. η χρονική πολυπλοκότητα καλύτερης περίπτωσης.
 - b. η χρονική πολυπλοκότητα μέσης περίπτωσης.
 - c. η χρονική πολυπλοκότητα χειρότερης περίπτωσης.
6. Ποιο είναι το άθροισμα της σειράς $1+2+3+\dots+n$;
7. Ποιο είναι το αποτέλεσμα της πράξης $\log_2 1048576 + \log_{10} 10000 + \log_3 27$;

8. Δίνεται η ακολουθία 9 τιμών: -2, 1, -3, 4, -1, 2, 1, -5, 4.
 - a. Ποια θα είναι η προθεματική ακολουθία (10 τιμές) που θα σχηματιστεί για την επίλυση του προβλήματος της μέγιστης υποακολουθίας (MAX SUBARRAY) με αλγόριθμο πολυπλοκότητας $O(n^2)$;
 - b. Ποια θα είναι η επιθεματική ακολουθία (10 τιμές) που θα σχηματιστεί για την επίλυση του προβλήματος της μέγιστης υποακολουθίας (MAX SUBARRAY) με τον αλγόριθμο του Kadane;
9. Στην υλοποίηση μιας επεκτάσιμης ακολουθίας ποιο είναι το πλεόν συμφέρον από τα ακόλουθα;
 - a. Όταν γεμίζει η ακολουθία να αντιγράφονται τα περιεχόμενά της σε νέα ακολουθία με διπλάσιο μέγεθος.
 - b. Όταν γεμίζει η ακολουθία να αντιγράφονται τα περιεχόμενά της σε νέα ακολουθία με μέγεθος μεγαλύτερο κατά 1 θέση
 - c. Όταν γεμίζει η ακολουθία να αντιγράφονται τα περιεχόμενά της σε ακολουθία με μέγεθος μεγαλύτερο κατά 100 θέσεις
10. Γράψτε κώδικα (η ψευδοκώδικα) που να αντιστρέφει μια ακολουθία τιμών χρησιμοποιώντας χώρο $O(1)$.
11. Η επιμερισμένη ανάλυση (amortized analysis) αφορά:
 - a. τον υπολογισμό της χρονικής πολυπλοκότητας μιας λειτουργίας.
 - b. τον υπολογισμό της χρονικής πολυπλοκότητας ενός συνόλου λειτουργιών.
 - c. το χρόνο που απαιτεί η εκτέλεση η εκτέλεση μιας λειτουργίας.
 - d. το χρόνο που απαιτεί η εκτέλεση ενός συνόλου λειτουργιών.

Κεφάλαιο 5

1. Απαντήστε με ΣΩΣΤΟ/ΛΑΘΟΣ
 - a. Η έννοια της ουράς προτεραιότητας και του σωρού είναι ταυτόσημες.
 - b. Ένας σωρός μπορεί να έχει διπλότυπα.
 - c. Η δημιουργία ενός σωρού με διαδοχικές εισαγωγές τιμών είναι ταχύτερη από τη δημιουργία ενός σωρού από μια ακολουθία με heapify.
 - d. Η ουρά προτεραιότητας είναι ένας αφηρημένος τύπος δεδομένων.
 - e. Ένας σωρός ελαχίστων είναι ένας αφηρημένος τύπος δεδομένων.
 - f. Η αποδοτικότερη υλοποίηση ενός σωρού είναι με δένδρα όπου κάθε κόμβος έχει 2 δείκτες προς τα παιδιά του.
 - g. Σε έναν σωρό μεγίστων ένας κόμβος έχει μικρότερη τιμή από τα παιδιά του.
 - h. Σε έναν σωρό ελαχίστων το ελάχιστο στοιχείο είναι τοποθετημένο στο χαμηλότερο επίπεδο του δένδρου στον πλέον δεξί κόμβο.
 - i. Η λίστα [2, 4, 6, 10, 8, 15] μπορεί να αντιπροσωπεύει έναν σωρό ελαχίστων;
2. Ποιες είναι οι χρονικές πολυπλοκότητες των ακόλουθων λειτουργιών σε έναν σωρό ελαχίστων;
 - a. Λήψη μικρότερης τιμής.
 - b. Αφαίρεση μικρότερης τιμής.
 - c. Εισαγωγή νέας τιμής.
 - d. heapify.
 - e. heapsort.
3. Τι θα εμφανίσει ο ακόλουθος κώδικας σε Python;

```
import heapq
li = [5, 7, 9, 1, 3]
li_minus = [-n for n in li]
heapq.heapify(li_minus)
print(-1 * heapq.heappop(li_minus))
```

Παρατήρηση: heapq είναι ένας σωρός ελαχίστων στην Python

4. Σε έναν σωρό μεγίστων εισάγονται σταδιακά οι τιμές: 5,12,7,9,16,11. Μετά από 3 εξαγωγές της μέγιστης τιμής ποια θα είναι τα περιεχόμενα του σωρού (να γράψετε τα στοιχεία με την σειρά που θα εμφανίζονται στον πίνακα που υποστηρίζει το σωρό);
5. Σε έναν σωρό σε ποιες θέσεις του πίνακα που τον υποστηρίζει εντοπίζονται τα παιδιά (αριστερό παιδί, δεξιό παιδί) του στοιχείου που βρίσκεται στη θέση 52. Σε ποια θέση βρίσκεται ο γονέας του ίδιου στοιχείου;

6. Ποια είναι η πλέον αποδοτική υλοποίηση μιας ουράς προτεραιότητας;
 - a. με σωρό
 - b. με αταξινόμητη λίστα
 - c. με ταξινομημένη λίστα
 - d. είναι το ίδιο και για τις 3 προηγούμενες περιπτώσεις
7. Γράψτε κώδικα (ή ψευδοκώδικα) που να εμφανίζει τις 10 μικρότερες τιμές από 1000 τιμές που θα εισάγονται σε έναν σωρό ελαχίστων.

Κεφάλαιο 6

1. Απαντήστε με ΣΩΣΤΟ/ΛΑΘΟΣ
 - a. Ο κατακερματισμός κλειστής διευθυνσιοδότησης μπορεί να έχει load factor πάνω από 100%.
 - b. Ο κατακερματισμός ανοικτής διευθυνσιοδότησης μπορεί να έχει load factor πάνω από 100%.
 - c. Μια συνάρτηση κατακερματισμού θα πρέπει για δύο διαφορετικές εισόδους να επιστρέφει την ίδια τιμή.
 - d. Με ευκολία και αποδοτικά μπορούμε να διασχίσουμε ταξινομημένα τα κλειδιά που βρίσκονται σε έναν πίνακα κατακερματισμού.
 - e. Οι κρυπτογραφικές συναρτήσεις κατακερματισμού χρησιμοποιούνται στην υλοποίηση πινάκων κατακερματισμού.
 - f. Ο διπλός κατακερματισμός είναι μια επέκταση του κατακερματισμού κλειστής διευθυνσιοδότησης (κατακερματισμός με αλυσίδες) για τον χειρισμό των συγκρούσεων.
 - g. Ο κατακερματισμός έχει εφαρμογή στην κρυπτογραφία.
 - h. Οι πίνακες κατακερματισμού συνήθως έχουν ως μέγεθος έναν πρώτο αριθμό.
 - i. Η συνάρτηση κατακερματισμού murmur είναι κρυπτογραφική.
 - j. Η συνάρτηση κατακερματισμού SHA256 είναι κρυπτογραφική.
2. Στον κατακερματισμό κούκου:
 - a. Υπάρχουν δύο πίνακες κατακερματισμού όπου σε κάθε θέση τους υπάρχει μία συνδεδεμένη λίστα.
 - b. Υπάρχει ένας πίνακα κατακερματισμού όπου σε κάθε θέση του υπάρχει μία συνδεδεμένη λίστα.
 - c. Υπάρχει ένας πίνακα κατακερματισμού όπου σε κάθε θέση του υπάρχει ένα στοιχείο.
 - d. Υπάρχουν δύο πίνακες κατακερματισμού όπου σε κάθε θέση τους υπάρχει ένα στοιχείο.
3. Σε έναν πίνακα κατακερματισμού με ανοικτή διευθυνσιοδότηση με χωρητικότητα 101 τιμών έχουν εισαχθεί 20 τιμές. Ποιο είναι το load factor;
4. Συμπληρώστε τις πολυπλοκότητες χρόνου για τις ακόλουθες περιπτώσεις:
 - a. Χρόνος χειρότερης περίπτωσης και επιμερισμένος χρόνος λειτουργίας εισαγωγής σε πίνακα κατακερματισμού ανοικτής διευθυνσιοδότησης
 - b. Χρόνος χειρότερης περίπτωσης και επιμερισμένος χρόνος λειτουργίας διαγραφής σε πίνακα κατακερματισμού ανοικτής διευθυνσιοδότησης
 - c. Χρόνος χειρότερης περίπτωσης και επιμερισμένος χρόνος λειτουργίας εύρεσης σε πίνακα κατακερματισμού ανοικτής διευθυνσιοδότησης
 - d. Χρόνος χειρότερης περίπτωσης και επιμερισμένος χρόνος λειτουργίας εισαγωγής σε πίνακα κατακερματισμού κούκου
 - e. Χρόνος χειρότερης περίπτωσης και επιμερισμένος χρόνος λειτουργίας διαγραφής σε πίνακα κατακερματισμού κούκου
 - f. Χρόνος χειρότερης περίπτωσης και επιμερισμένος χρόνος λειτουργίας εύρεσης σε πίνακα κατακερματισμού κούκου
5. Δίνεται μια λίστα με διευθύνσεις IP επισκεπτών που επισκέφθηκαν μια ιστοσελίδα. Γράψτε αποδοτικό κώδικα (ή ψευδοκώδικα) που να εντοπίζει το πλήθος των μοναδικών επισκέψεων.
6. Δίνονται 2 ακολουθίες, η ακολουθία A με M στοιχεία και η ακολουθία B με N στοιχεία αντίστοιχα. Θεωρείστε ότι το M είναι σημαντικά μικρότερο από το N. Γράψτε κώδικα (ή ψευδοκώδικα) που να εξετάζει αποδοτικά το εάν οι ακολουθίες είναι ξένες μεταξύ τους. Τι υπολογιστική πολυπλοκότητα θα έχει ο αλγόριθμός σας;

7. Γράψτε κώδικα σε Python στον οποίο ο χρήστης θα δίνει ονόματα χρηστών. Αν ο χρήστης υπάρχει θα επιστρέφει το password του. Αν δεν υπάρχει θα τον αποθηκεύει και θα ζητάει από τον χρήστη να εισάγει το password του.

Κεφάλαιο 7

1. Απαντήστε με ΣΩΣΤΟ/ΛΑΘΟΣ
 - a. Η υλοποίηση της δομής ένωσης-εύρεσης με συνδεδεμένες λίστες είναι αποδοτικότερη από την υλοποίηση της με δένδρα.
 - b. Η υλοποίηση της δομής-ένωσης εύρεσης με δένδρα στηρίζεται σε δομή δένδρων με δείκτες από τον κάθε κόμβο γονέα προς τα παιδιά του.
 - c. Η δομή ένωσης-εύρεσης είναι γνωστή και ως δομή ξένων συνόλων.
 - d. Η συμπίεση μονοπατιού είναι μια τεχνική που εφαρμόζεται κατά την εισαγωγή νέων τιμών σε μια δομή ένωσης-εύρεσης.
 - e. Στη δομή ένωσης-εύρεσης στην ένωση κατά μέγεθος (union by size), το σύνολο με τα περισσότερα στοιχεία προσαρτάται ως παιδί στο αντιπροσωπευτικό στοιχείο του συνόλου με τα λιγότερα στοιχεία.
2. Ποιες είναι οι λειτουργίες που υποστηρίζει μια δομή ένωσης-εύρεσης;
3. Έστω ο αλγόριθμος δημιουργίας λαβυρίνθων με δομή ένωσης-εύρεσης. Ποια θα είναι η ενδιάμεση (μη τελική) μορφή του ακόλουθου λαβυρίνθου αν επιλεγούν με τη σειρά οι τοίχοι 3-6, 2-3, 5-6 και 2-5. Ποια θα είναι τα ξένα σύνολα που θα έχουν δημιουργηθεί;

1	2	3
4	5	6

4. Δίνεται ένα σύνολο από σχέσεις φιλίας ανάμεσα σε άτομα (κάθε άτομο αναπαρίσταται από έναν αριθμό και η σχέση φιλίας με ένα ζεύγος αριθμών). Γράψτε Python κώδικα που να εντοπίζει τις ομάδες ατόμων που είναι συνδεδεμένες χρησιμοποιώντας δομή ένωσης εύρεσης (π.χ. friends = [[1, 0], [2, 3], [1, 2], [4, 5]] με ομάδες [0,1,2,3] και [4,5]). Θεωρείστε ότι μπορείτε να χρησιμοποιήσετε τη βιβλιοθήκη της Python unionfind με την ακόλουθη λειτουργικότητα:

```
from unionfind import unionfind

u = unionfind(3) # There are 3 items.
u.unite(0, 2) # Set 0 and 2 to same group.
u.issame(1, 2) # Ask "Are 1 and 2 same?"
u.groups() # Return groups.
```

Κεφάλαιο 8

1. Απαντήστε με ΣΩΣΤΟ/ΛΑΘΟΣ
 - a. Το κάτω όριο για αλγόριθμους ταξινόμησης με συγκρίσεις είναι $\Omega(n \log n)$.
 - b. Το κάτω όριο για αλγόριθμους ταξινόμησης με συγκρίσεις είναι $O(n \log n)$.
 - c. Το κάτω όριο για αλγόριθμους ταξινόμησης με συγκρίσεις είναι $O(n^2)$.
 - d. Ο αλγόριθμος γρήγορης ταξινόμησης συνήθως είναι ταχύτερος από τον αλγόριθμο ταξινόμησης με συγχώνευση και προτιμάται για ταξινόμηση πολλών δεδομένων που βρίσκονται στην κύρια μνήμη.
 - e. Ο αλγόριθμος ταξινόμησης με συγχώνευση είναι τυχαιοποιημένος αλγόριθμος.
 - f. Ένας αλγόριθμος σταθερής ταξινόμησης (stable sort) επιτρέπει σε μια λίστα με φοιτητές που ταξινομείται πρώτα κατά όνομα και μετά ταξινομείται ξανά κατά έτος να εμφανίζει τους φοιτητές του κάθε έτους ταξινομημένους σε αλφαβητική σειρά ονόματος.
 - g. Η MergeSort περιέχει δύο αναδρομικές κλήσεις της.
 - h. Η BinarySearch περιέχει δύο αναδρομικές κλήσεις της.
 - i. Η εκτέλεση της QuickSort για μια ακολουθία n στοιχείων, προκαλεί αναδρομικές κλήσεις της QuickSort για 2 ακολουθίες μεγέθους $n/2$ η κάθε μια.

2. Συμπληρώστε τις πολυπλοκότητες χρόνου για τις ακόλουθες περιπτώσεις;
 - a. Πολυπλοκότητα χρόνου χειρότερης περίπτωσης για τον αλγόριθμο QuickSort.
 - b. Πολυπλοκότητα χρόνου μέσης περίπτωσης για τον αλγόριθμο QuickSort.
 - c. Πολυπλοκότητα χρόνου χειρότερης περίπτωσης για τον αλγόριθμο MergeSort
 - d. Πολυπλοκότητα χρόνου χειρότερης περίπτωσης για τον αλγόριθμο InsertionSort
3. Δείξτε τη σταδιακή ταξινόμηση με MergeSort της ακολουθίας 15,7,2,8,19,11,3,6,17.
4. Δείξτε τη σταδιακή ταξινόμηση με QuickSort της ακολουθίας 15,7,2,8,19,11,3,6,17 θεωρώντας ότι ο διαμερισμός γίνεται κάθε φορά διατηρώντας τη σχετική θέση μεταξύ των στοιχείων που είναι μικρότερα του ρινοτ και τη σχετική θέση μεταξύ των στοιχείων που είναι μεγαλύτερα του ρινοτ.
5. Δίνεται η ακολουθία τιμών 15,7,2,8,19,11,3,6,17. Ποια θα είναι η μορφή της ακολουθίας αν εφαρμοστεί διαμερισμός με ρινοτ το τελευταίο στοιχείο της ακολουθίας;
 - a. χρησιμοποιώντας βοηθητικό πίνακα
 - b. in-place.
6. Γράψτε κώδικα (ή ψευδοκώδικα) συγχώνευσης 2 ταξινομημένων σε αύξουσα σειρά πινάκων με N τιμές ο καθένας, που να επιστρέφει τις N μικρότερες τιμές και από τους 2 πίνακες. Ο κώδικας να έχει χρονική πολυπλοκότητα $O(N)$.
7. Γράψτε κώδικα (ή ψευδοκώδικα) που να πραγματοποιεί διαμερισμό σε έναν πίνακα, δεχόμενος ως είσοδο τον πίνακα και τη θέση του ρινοτ. Να χρησιμοποιηθεί βοηθητικός πίνακας.

Κεφάλαιο 10

1. Απαντήστε με ΣΩΣΤΟ/ΛΑΘΟΣ
 - a. Η άπληστη μέθοδος επιστρέφει τη βέλτιστη λύση για το κλασματικό πρόβλημα σακιδίου.
 - b. Η άπληστη μέθοδος επιστρέφει τη βέλτιστη λύση για το 0-1 πρόβλημα σακιδίου.
 - c. Η άπληστη μέθοδος επιστρέφει τη βέλτιστη λύση για το πρόβλημα χρονοπρογραμματισμού εργασιών σε πολλαπλές ίδιες μηχανές με γνωστές ώρες έναρξης και τερματισμού της κάθε εργασίας, εφόσον οι εργασίες ανατεθούν στις μηχανές με σειρά μεγέθους.
 - d. Η άπληστη μέθοδος επιστρέφει τη βέλτιστη λύση για το πρόβλημα χρονοπρογραμματισμού εργασιών σε πολλαπλές μηχανές με γνωστές ώρες έναρξης και τερματισμού της κάθε εργασίας, εφόσον οι εργασίες ανατεθούν στις μηχανές με σειρά ώρας έναρξης.
 - e. Η άπληστη μέθοδος συνήθως λύνει βέλτιστα τα προβλήματα.
 - f. Η κωδικοποίηση Huffman χρησιμοποιεί το ίδιο πλήθος δυαδικών ψηφίων για να κωδικοποιήσει κάθε χαρακτήρα του κειμένου.
 - g. Η κωδικοποίηση Huffman είναι ένα είδος κρυπτογράφησης.
 - h. Η κωδικοποίηση Huffman μπορεί να θεωρηθεί ως μια μορφή συμπίεσης.
2. Δίνονται τα ακόλουθα αντικείμενα τα οποία συμμετέχουν στο πρόβλημα του κλασματικού σακιδίου. Αντικείμενο A με βάρος 100kg και αξία 20 ευρώ, αντικείμενο B με βάρος 300kg και αξία 150 ευρώ, αντικείμενο Γ με βάρος 5kg και αξία 10 ευρώ. Με δεδομένο ότι το επιτρεπτό βάρος του σακιδίου είναι 150kg να υπολογιστεί η βέλτιστη λύση του προβλήματος.
3. Δίνεται μια λίστα αντικειμένων για τα οποία γνωρίζουμε την ποσότητά του κάθε αντικειμένου σε κιλά και την αξία του σε ευρώ. Αν θεωρήσουμε ότι μπορούμε να επιλέξουμε για κάθε αντικείμενο ένα αυθαίρετα μικρό ή μεγάλο ποσοστό της διαθέσιμης ποσότητας του, να επιλεγούν τα αντικείμενα που θα δώσουν τη μεγαλύτερη αξία δεδομένης της συνολικής ποσότητας σε κιλά που μπορεί να αποκτηθεί. Γράψτε κώδικα (ή ψευδοκώδικα) που να υλοποιεί τα παραπάνω.
4. Δίνεται μια λίστα εργασιών με γνωστούς χρόνους έναρξης και τερματισμού. Κάθε εργασία μπορεί να εκτελεστεί σε μια μηχανή και υπάρχουν πολλές διαθέσιμες ίδιες μηχανές. Γράψτε κώδικα (ή ψευδοκώδικα) που να αναθέτει βέλτιστα τις εργασίες σε μηχανές έτσι ώστε να χρησιμοποιούνται συνολικά οι λιγότερες δυνατές μηχανές.
5. Δίνεται η ακόλουθη φράση «KEEP CALM AND CARRY ON». Χρησιμοποιήστε την κωδικοποίηση Huffman για την κωδικοποίηση της.