

Αλγόριθμοι και Πολυπλοκότητα – ενδιάμεσες εξετάσεις χειμερινού εξαμήνου 2023-2024

Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων

4/12/2023

Διάρκεια εξέτασης: 2 ώρες

Θέμα 1 [1, 1]

A) Απαντήστε με ΣΩΣΤΟ ή ΛΑΘΟΣ:

1. Για το πρόβλημα της μέγιστης υποακολουθίας (MAXSUBARRAY) μπορεί να υπάρξει αλγόριθμος που να το επιλύει σε χρόνο καλύτερο από $O(n)$.
2. Ένας αλγόριθμος με πολυπλοκότητα $O(\log n)$ γενικά προτιμάται έναντι ενός άλλου αλγορίθμου που επιλύει το ίδιο πρόβλημα και είναι πολυπλοκότητας $O(n)$.
3. Η συνάρτηση $2n \log n + 5(n+2)^2 + 7(n-100)^3 + 16n + 10^5 + 1000 \log n$ είναι $O(n^2)$.
4. Η επιμερισμένη ανάλυση μιας επεκτάσιμης ακολουθίας αποδεικνύει ότι είναι προτιμότερο όταν επιχειρείται εισαγωγή σε μια ήδη γεμάτη ακολουθία να αντιγράφονται τα περιεχόμενά της σε μια διπλάσιου μεγέθους ακολουθία έναντι της αντιγραφής των περιεχομένων της σε μια ακολουθία με μεγαλύτερο μέγεθος κατά ένα σταθερό αριθμό θέσεων (π.χ. 10 θέσεις).
5. Η υλοποίηση της επεκτάσιμης ακολουθίας γίνεται στην Python με τη δομή δεδομένων tuple.

B) Για το πρόβλημα της μέγιστης υποακολουθίας (MAXSUBARRAY), δίνεται η ακολουθία 7 τιμών:
-4, 5, 7, -6, 10, -15, 3

- i) Συμπληρώστε στον ακόλουθο πίνακα, τα προθεματικά και τα επιθεματικά αθροίσματα.

Ακολουθία τιμών		-4	5	7	-6	10	-15	3
Προθεματικά αθροίσματα								
Επιθεματικά αθροίσματα								
Δείκτες	0	1	2	3	4	5	6	7

- ii) Πως προκύπτει από τα προθεματικά αθροίσματα, η μέγιστη υποακολουθία 5, 7, -6, 10 που έχει άθροισμα 16 στο συγκεκριμένο παράδειγμα;

Θέμα 2 [1, 1]

A) Ποιες είναι οι 3 βασικές λειτουργίες του αφηρημένου τύπου δεδομένων «Ουρά Προτεραιότητας» και ποια είναι η πολυπλοκότητα της κάθε λειτουργίας, αν υλοποιηθεί με τη δομή δεδομένων «Σωρός μεγίστων»;

B) Δίνεται ο ακόλουθος κώδικας σε Python. Συμπληρώστε τη συνάρτηση ταξινόμησης σωρού (heapsort) έτσι ώστε να δέχεται μια λίστα τιμών και να επιστρέφει μια ταξινομημένη λίστα.

```
# παραδείγματα με λειτουργίες της heapq
# nums = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
# heapq.heapify(nums) # μετατροπή λίστας nums σε σωρό
# heapq.heappush(nums, 7) # προσθήκη 7 στο σωρό
# min_val = heapq.heappop(nums) # αφαίρεση και επιστροφή της μικρότερης τιμής του σωρού
import heapq
```

```
def heapsort(nums):  
    pass  
  
nums = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]  
nums = heapsort(nums)  
print(nums)
```

Θέμα 3 [1, 2]

A) Γράψτε αποδοτικό κώδικα (ή ψευδοκώδικα) που να δέχεται μια λίστα τιμών χρωμάτων ως τριάδες τιμών RGB (Red Green Blue) από το 0 μέχρι το 255 και να επιστρέφει το πλήθος των διαφορετικών χρωμάτων.

B) Δίνεται ένας αριθμός πιστωτικών καρτών (π.χ. 20.000) και ένας αριθμός συναλλαγών (π.χ. 10.000.000) που η κάθε μια περιέχει έναν αριθμό κάρτα και ένα ποσό συναλλαγής (π.χ. 1234-5678-9012-3456, 20 ευρώ).

- i) Περιγράψτε **λεκτικά** έναν αποδοτικό τρόπο που μπορεί να εφαρμοστεί για να υπολογιστούν οι 10 κάρτες με το συνολικά υψηλότερο ποσό συναλλαγών.
- ii) Ποια θα ήταν η προσέγγισή σας αν έπρεπε να χρησιμοποιήσετε τη γλώσσα προγραμματισμού C και δεν είχατε πρόσβαση σε εξωτερικές βιβλιοθήκες της γλώσσας;

Θέμα 4 [1, 1, 1]

A) 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 1-2, 5-8, 7-8, 4-7 και 4-5. Ποια θα είναι τα ξένα σύνολα που θα έχουν δημιουργηθεί; Σχεδιάστε το λαβύρινθο μετά τις παραπάνω ενέργειες.

1	2	3
4	5	6
7	8	9

B) Δείξτε τη σταδιακή ταξινόμηση εφαρμόζοντας την ταξινόμηση με συγχώνευση (MergeSort) της ακολουθίας τιμών 5,2,3,8,1,9,7,4,0,6. Ποια είναι η πολυπλοκότητα καλύτερης και ποια η πολυπλοκότητα χειρότερης περίπτωσης της MergeSort; Αναφέρατε 3 ομοιότητες της MergeSort με τη γρήγορη ταξινόμηση (QuickSort).

Γ) Γράψτε κώδικα (ή ψευδοκώδικα) που να πραγματοποιεί διαμερισμό (partition) των στοιχείων ενός πίνακα A, N θέσεων με βάση το πρώτο στοιχείο του, χρησιμοποιώντας βοηθητικό πίνακα.