# Αλγόριθμοι και Πολυπλοκότητα – ενδιάμεσες εξετάσεις χειμερινού εξαμήνου 2024-2025

Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων 5/12/2024

Διάρκεια εξέτασης: 2 ώρες

## Θέμα 1 [A=1, B=2]

Α) Απαντήστε με ΣΩΣΤΟ ή ΛΑΘΟΣ:

- 1. Η πολυπλοκότητα χειρότερης περίπτωσης της QuickSort είναι O(n log n).
- 2. Ο κατακερματισμός κούκου χρησιμοποιεί 2 συναρτήσεις κατακερματισμού.
- 3. Ο συντελεστής φόρτωσης (load factor) ενός πίνακα κατακερματισμού ισούται με το μέγεθος του πίνακα κατακερματισμού διά του πλήθους των στοιχείων που έχουν εισαχθεί στον πίνακα κατακερματισμού.
- 4. Η συνάρτηση κατακερματισμού murmur είναι κρυπτογραφική συνάρτηση κατακερματισμού.
- 5. Στον κατακερματισμό κούκου η λειτουργία της εύρεσης ενός κλειδιού και η λειτουργία της διαγραφής ενός κλειδιού είναι πάντα O(1).

B) Δίνεται η ακόλουθη αναπαράσταση, σε Python, ενός κατευθυνόμενου γράφου ως λίστα γειτονικότητας (adjacency list):

from collections import defaultdict graph = defaultdict(list, {"a":["b","c","f"], "b":["c","d","o"]

Πίνακας 1

Καταγράψτε το παραπάνω γράφημα ως πίνακα γειτονικότητας στον Πίνακα 1. Γράψτε κώδικα που να υλοποιεί τη συνάρτηση is\_adjacent(graph, v1, v2) και που να επιστρέφει True αν υπάρχει ακμή από την κορυφή v1 προς την κορυφή v2, αλλιώς να επιστρέφει False.

## Θέμα 2 [A=1, B=1]

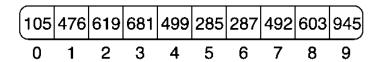
A) Δίνεται μια λίστα η τιμών. Γράψτε κώδικα σε Python (ή ψευδοκώδικα) που να υπολογίζει σε γραμμικό χρόνο την τιμή που εμφανίζεται περισσότερες φορές από το μισό του πλήθους των στοιχείων της λίστας. Υποθέστε ότι υπάρχει πάντα μια τέτοια τιμή στα δεδομένα. Για παράδειγμα για τη λίστα τιμών [2,2,1,2,3,1,2,2] η τιμή που ζητείται είναι η τιμή 2.

B) Προτείνετε λύση που να επιλύει το πρόβλημα του προηγούμενου ερωτήματος σε γραμμικό χρόνο χωρίς να χρησιμοποιηθεί επιπλέον δομή αποθήκευσης ενδιάμεσων τιμών, πέρα από την αρχική ακολουθία τιμών.

#### Θέμα 3 [A=1, B=1, Γ=1]

A) Δείξτε, σχεδιάζοντας μια δενδρική δομή, τη σταδιακή ταξινόμηση που επιτελείται εφαρμόζοντας τη γρήγορη ταξινόμηση (QuickSort) για την ακολουθία τιμών 7, 0, 9, 1, 4, 5, 2, 3, 8, 6 όπου κάθε φορά επιλέγεται ως pivot το πλέον αριστερό στοιχείο της υποακολουθίας.

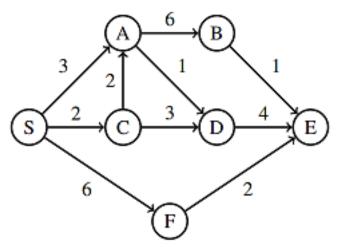
B) Για τον διαμερισμό in-place της QuickSort γράψτε την τελική μορφή της ακολουθίας που παρουσιάζεται στη συνέχεια, χρησιμοποιώντας ως pivot τον αριθμό 499.



Γ) Ποια είναι η καλύτερη πολυπλοκότητα που μπορεί να επιτευχθεί σε αλγόριθμο ταξινόμησης που στηρίζεται σε συγκρίσεις; Αιτιολογήστε γιατί.

### Θέμα 4 [A=1, B=1]

Δίνεται το ακόλουθο κατευθυνόμενο γράφημα:



A) Εφαρμόστε τον αλγόριθμο εύρεσης των συντομότερων διαδρομών του Dijkstra με αφετηρία την κορυφή S και συμπληρώστε από αριστερά προς τα δεξιά όλες τις τιμές που διαδοχικά λαμβάνει ο ακόλουθος πίνακας των ελάχιστων αποστάσεων από την αφετηρία. Με ποια σειρά εντάσσονται οι κορυφές στις «λυμένες» κορυφές, δηλαδή στις κορυφές για τις οποίες έχει βρεθεί η μικρότερη απόσταση από την αφετηρία; Ποια είναι η χρονική πολυπλοκότητα του αλγορίθμου;

S	Α	В	С	D	E	F
0	+∞	+∞	*8	+8	+∞	+∞

B) Εφαρμόστε την πρώτη μόνο επανάληψη (1<sup>st</sup> iteration) του αλγορίθμου εύρεσης των συντομότερων διαδρομών των Bellman Ford, με αφετηρία την κορυφή S, εξετάζοντας τις ακμές με την ακόλουθη σειρά (SA, SC, SF, AB, AD, BE, CA, CD, DE, FE). Συμπληρώστε τις τιμές αποστάσεων που λαμβάνουν σταδιακά όλες οι κορυφές. Ποια είναι η χρονική πολυπλοκότητα του αλγορίθμου;

S	Α	В	С	D	E	F
0	+∞	+∞	+∞	+∞	8+	+∞