352. ΔΟΜΕΣ ΔΕΔΟΜΈΝΩΝ

Εργασία Εξαμήνου

Εαρινό Εξάμηνο 2024

Γιάννης Λιβιεράτος
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Σχολή Θετικών Επιστημών, Τμήμα Μαθηματικών

Παράδοση: 25 Ιουνίου 2024

Contents

0 Οδηγίες

Η προθεσμία της παρούσας εργασίας είναι η μέρα και ώρα της εξέτασης: **25 Ιουνίου, 18:00**. Μπορεί να αποσταλεί με οποιονδήποτε τρόπο, αλλά προτείνεται η e-class και το e-mail σε περίπτωση που γραφτεί σε υπολογιστή (word, latex,...) και η παράδοση στο γραφείο 306, **με ειδοποίηση μέσω e-mail ότι αφέθηκε εκεί αν δεν μου δοθεί στο χέρι** σε περίπτωση που είναι χειρόγραφη. Στον χειρότερο συνδυασμό, που είναι φωτογραφίες χειρόγραφου, φροντίστε όσο μπορείτε να είναι καθαρά τα γράμματα και καλή η ανάλυση της φωτογραφίας.

Επίσης πολύ θα χαρώ να λάβω και προγραμματάκια σε όποια γλώσσα θέλετε αντί για αλγορίθμους σε ψευδογλώσσα. Αν θέλετε να στείλετε κώδικα που τρέχει, καλό θα είναι να τον έχετε (και) σε .txt αρχείο για να μην υπάρξει θέμα με ασυμβατότητες (χρησιμοποιώ linux) και να είναι και επαρκώς ή υπερβολικά σχολιασμένος. Θερμή παράκληση, αν θέλετε να μου στείλετε κώδικα που **δεν** τρέχει, να μου έχετε κάπως γράψει που σκάει και τι υποθέτεται ότι φταίει (ή γιατί θεωρείτε ότι θα πρεπε να μην σκάει).

Προτείνω να γράφετε ερωτήσεις, είτε σε εμένα είτε στο forum της e-class, στο βαθμό που αυτές είναι συγκεκριμένες και για επιμέρους θέματα. Δηλαδή όχι "πως λύνεται η X άσκηση;" αλλά "γιατί δεν μπορούμε να χρησιμοποιήσουμε την Y μέθοδο για την X άσκηση όπως βλέπω στην Z ιστοσελίδα;". Προτείνω ισχυρά την αναγραφή πηγών αν χρησιμοποιήσετε βιβλία, άλλες σημειώσεις, λύσεις συμφοιτητ(ρι)ών σας, online ιστοσελίδες ή εργαλεία τύπου ChatGPT. Σε περίπτωση που παρθεί έτοιμη λύση από κάπου, προτείνω επίσης ισχυρά να το δηλώσετε και να εξηγήσετε γιατί δουλεύει αυτό που βρήκατε, με δικό σας τρόπο.

Σε περίπτωση που λύσετε ασκήσεις σε ομάδες, ας αναγράφεται στην αρχή του αρχείου/εντύπου που θα παραδώσετε. Θεωρώ λογική μία ομάδα έως και 4ων ατόμων, μεγάλη μία που φτάνει τα 6 και "ξεχείλωμα" οποιονδήποτε μεγαλύτερο αριθμό ατόμων σε ομάδα.

Για όσα άτομα επιθυμείτε να παρουσιάσετε, η άσκηση που θα αναλάβετε για παρουσίαση θα είναι καλό να μου έχει σταλεί έστω και απομονωμένα τουλάχιστον μία μέρα πριν την παρουσίαση, ώστε να σας έχω προειδοποιήσει για τυχόν λάθη στις λύσεις σας.

Η εργασία προσφέρει **έως και 3 μονάδες** στον τελικό βαθμό. Για να πάρετε όλες τις μονάδες, καλό είναι να παραδώσετε γύρω στις 10 ασκήσεις. Θα λάβω υπόψιν σε αυτό τις παρουσιάσεις, το αν λύσετε περισσότερες από 10, μη ολοκληρωμένες ασκήσεις, την προσπάθεια ακόμη και σε λάθος απαντήσεις και την όποια διαπραγμάτευση επιθυμείτε αφού σας ανακοινώσω βαθμούς.

Για όσα από εσάς κάνετε αριθμητική ελάχιστου κόπου για να περάσετε το μάθημα, θα θέλει λιγότερο κόπο να πάρετε 5 στην τελική εξέταση, παρά 3 μονάδες από την εργασία και 2 στην τελική εξέταση. **Δεν** θα γίνει καμία απόπειρα από πλευράς μου να ελέγξω "αντιγραφές" και τα λοιπά.

1 Εισαγωγικά

Άσκηση 1. Στις παρακάτω περιπτώσεις, εξετάστε αν $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$ ή $f(n) = \Theta(g(n))$. Bonus: Αν δεν ισχύει $f(n) = \Theta(g(n))$, εξετάστε αν $f(n) = o(g(n))$ ή αν $f(n) = \omega(g(n))$.

(i) $f(n) = n - 100$, $g(n) = n - 200$.

(ii) $f(n) = n^{1/2}$, $g(n) = n^{2/3}$.

(iii) $f(n) = 100n + \log(n)$, $g(n) = n + (\log(n))^2$.

(iv) $f(n) = n\log(n)$, $g(n) = 10n\log(10n)$.

(v) $f(n) = \log(2n)$, $g(n) = \log(3n)$.

(vi) $f(n) = 10\log(n)$, $g(n) = \log(n^2)$.

(vii) $f(n) = n^{1.01}$, $g(n) = n\log^2(n)$.

(viii) $f(n) = n^2/\log(n)$, $g(n) = n(\log(n))^2$.

(ix) $f(n) = n^{0.1}$, $g(n) = (\log(n)^1 0$.

(x) $f(n) = \log(n)^{\log(n)}$, $g(n) = n/\log(n)$.

(xi) $f(n) = \sqrt{n}$, $g(n) = (\log(n))^3$.

(xii) $f(n) = n^{1/2}$, $g(n) = 5^{\log(n)}$.

(xiii) $f(n) = n2^n$, $g(n) = 3^n$.

(xiv) $f(n) = 2^n$, $g(n) = 2^{n+1}$

(xv) $f(n) = n!$, $g(n) = 2^n$.

(xvi) $f(n) = (\log(n))^{\log(n)}$, $g(n) = 2^{(\log(n))^2}$.

(xvii) $f(n) = \sum_{i=1}^{n} i^k$, $g(n) = n^{k+1}$.

Άσκηση 2. Έστω $f(n) = 1 + c + c^2 + \cdots + c^n$, όπου $n \in \mathbb{N}$ και $c \in \mathbb{R}$, $c > 0$. Να δείξετε ότι:

(i) αν $c < 1$, $f(n) = \Theta(1)$,

(ii) αν $c = 1$, $f(n) = \Theta(n)$ και

(iii) αν $c > 1$, $f(n) = \Theta(c^n)$.

Άσκηση 3. Έστω γράφημα $G = (V, E)$.

(i) Να βρεθεί άνω φράγμα στο πλήθος $|E|$ των ακμών που μπορεί να έχει και να αποδειχθεί η εγκυρότητά του.

(ii) Να υπολογισθεί το $\sum_{u \in V} \deg(u)$.

(iii) Να βρεθεί άνω φράγμα στο πλήθος των κύκλων που μπορεί να έχει.

Άσκηση 4. (i) Έστω ένα κατευθυνόμενο γράφημα $D = (G, E)$ και $S_1, \ldots, S_k$ οι ισχυρά συνεκτικές του συνιστώσες. Τι θα συμβεί σε αυτές, αν προστεθεί μία ακμή στο σύνολο E; Να βρεθεί άνω φράγμα στο πλήθος των ακμών που πρέπει να προστεθούν ώστε το D να έχει ακριβώς μία ισχυρά συνεκτική συνιστώσα.

(ii) Ένας *περίπατος Euler* σε ένα κατευθυνόμενο ισχυρά συνεκτικό γράφημα $D = (G, E)$, είναι μία ακολουθία, πιθανώς επαναλαμβανόμενων, κορυφών του D, έτσι ώστε κάθε ακμή του D να χρησιμοποιείται ακριβώς μία φορά. Να δείξετε ότι το D έχει τέτοιον περίπατο, αν και μόνον αν για κάθε $u \in V$, $\deg^{in}(u) = \deg^{out}(u)$.

2 Πίνακες-Λίστες

Άσκηση 5. Έστω πίνακας $n \times m$ πίνακας A, τέτοιος ώστε:

- $A[i][j] \leq A[i'][j]$ για κάθε $i < i'$ και $j = 1, \ldots, m$,
- $A[i][j] \leq A[i][j']$ για κάθε $j < j'$ και $i = 1, \ldots, n$.

Να γραφεί αλγόριθμος που εξετάζει ποια από τα στοιχεία $x_1, \ldots, x_k$ υπάρχουν στον A, χρόνου $O(k \log(nm))$. Αν για την απόδειξη του χρόνου χρειάζεστε το n, το m ή/και το nm να είναι κάποιας συγκεκριμένης μορφής, να δείξετε τι αλλαγή θα χρειαστεί να γίνει στον A και πόσο χρόνο θα χρειαστεί.

Bonus: Αν $x_1 \leq x_2 \leq \cdots \leq x_n$, μπορούμε να επιτύχουμε καλύτερο χρόνο; Είτε πραγματικό, είτε ασυμπτωτικό;

Άσκηση 6. Μία συμβολοσειρά $w = w_1 w_2 \cdots w_n$ ονομάζεται καρκινική, αν $w_i = w_{n-i+1}$, $i = 1, \ldots, n$.

(i) Περιγράψτε αλγόριθμο που κατασκευάζει μία απλή συνδεδεμένη λίστα, κάθε κόμβος της οποίας περιέχει κι ένα από τα σύμβολα της w.

(ii) Περιγράψτε αλγόριθμο που ελέγχει σε μια τέτοια λίστα, αν η w είναι καρκινική. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του.

(iii) Μπορείτε να βελτιώσετε τον αλγόριθμο σε περίπτωση που η λίστα είναι διπλή;

(iv) Μπορείτε να βελτιώσετε τον αλγόριθμο σε περίπτωση που η λίστα είναι κυκλική;

Άσκηση 7. Έστω L μια λίστα που περιέχει το αλφαριθμητικό w_i στον i-οστό της κόμβο, $i = 1, \ldots, n$.

(i) Περιγράψτε αλγόριθμο που αντιστρέφει την σειρά των αλφαριθμητικών, ώστε τώρα ο i-οστός της κόμβος να περιέχει το w_{n-i+1}, $i = 1, \ldots, n$.

(iii) Μπορείτε να βελτιώσετε τον αλγόριθμο σε περίπτωση που η λίστα είναι διπλή;

(iv) Μπορείτε να βελτιώσετε τον αλγόριθμο σε περίπτωση που η λίστα είναι κυκλική;

Άσκηση 8.

(i) Περιγράψτε αλγόριθμο που διαγράφει το $(\texttt{len}(L) - k)$-οστό στοιχείο μίας διπλής λίστας L. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του.

(ii) Περιγράψτε αλγόριθμο που διαγράφει το $(\texttt{len}(L) - k)$-οστό στοιχείο μίας απλής λίστας L. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του.

(iii) Περιγράψτε αλγόριθμο που διαγράφει το $(\texttt{len}(L) - k)$-οστό στοιχείο μίας κυκλικής λίστας L. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του.

Άσκηση 9.

(i) Περιγράψτε αλγόριθμο που δέχεται τις κεφαλές $_1$, $_2$ δύο απλών λιστών και επιστρέφει μία απλή λίστα που ξεκινάει με τους κόμβους της H_1 και τελειώνει με αυτούς της H_2. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του.

(ii) Περιγράψτε αλγόριθμο που παίρνει ως είσοδο μια κυκλική λίστα C ακεραίων και την επιστρέφει, προσθέτοντας i στο i-οστό της στοιχείο, $i = 1, \ldots, \texttt{len}(C)$. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του.

Άσκηση 10. Έστω μία απλή λίστα με ακέραιους αριθμούς. Περιγράψτε αλγόριθμο που ελέγχει αν το άθροισμα των αριθμών αυτών είναι μεγαλύτερο ή μικρότερο από το 0. Αν είναι μικρότερο, ο αλγόριθμος θα πρέπει να βρίσκει κατάλληλους κόμβους $u_1, \ldots, u_k$, ώστε αν αντιστρέψει τα πρόσιμά τους, το άθροισμα να γίνεται θετικό. Θέλουμε το k να είναι το ελάχιστο δυνατόν. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του αλγορίθμου σας.

Άσκηση 11. Έστω απλή λίστα L που περιέχει τους φυσικούς αριθμούς $n_1, \ldots, n_k \in \mathbb{N}$, $k \in \mathbb{N} \setminus \{0\}$. Να γραφεί αλγόριθμος που να διορθώνει την λίστα ώστε τελικά:

- ο αριθμός $n_i \in \mathbb{N}$ να βρίσκεται στον n_i-οστό κόμβο της L, $i = 1, \ldots, k$ και
- αν χρειάζεται, η λίστα να χρησιμοποιεί κόμβους με τιμές _ για να γεμίζει τα κενά.

Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του αλγορίθμου σας.

Μπορείτε να υποθέσετε ότι η λίστα σας είναι διπλή ή κυκλική. Αν το κάνετε, εξηγήστε σε τι σας διευκολύνει.

Άσκηση 12. Έστω λίστα L κόμβους $H = u_0, u_1, \ldots, u_n$:

- $u_i = (u_i.\texttt{prev}, u_i.\texttt{val}, u_i.\texttt{next})$, $i = 0, \ldots, n$,
- $H.\texttt{prev} = \texttt{NIL}$, $H.\texttt{val} = \emptyset$, $H.\texttt{next} = \texttt{addr}(u2)$,
- $u_{2i-1}.\texttt{prev} = u_{2i-1}.\texttt{next} = \texttt{NIL}$, $i = 1, \ldots, \lceil n/2 \rceil$,
- $u_{2i}.\texttt{prev} = u_{2i-1}$, $u_{2i}. = u_{2i+2}$.
- $u_i.\texttt{val} \in \mathbb{N}$, $i = 1, \ldots, n$.

Περιγράψτε αλγορίθμους για τις πέντε βασικές λειτουργίες: έλεγχος κενής δομής, προθήκη κόμβου, αφαίρεση κόμβου, αναζήτηση και προσπέλαση δομής.

Για την προσπέλαση, ελέγξτε αν οι τιμές στους κόμβους με άρτια απόσταση από την H είναι άρτιοι αριθμοί και οι τιμές σε αυτούς με περιττή απόσταση, περιττές. Αν δεν είναι, μηδενίστε τους. Διαγράψτε όλους τους μηδενισμένους κόμβους με περιττή απόσταση και μετρήστε πόσοι είναι οι εναπομείναντες μηδενισμένοι κόμβοι. Αν είναι γνησίως περισσότεροι από τους μη μηδενισμένους, αδειάστε την λίστα.

Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

3 Στοίβες-Ουρές

Άσκηση 13. Περιγράψτε αλγορίθμους για τις βασικές μεθόδους των στοιβών -έλεγχος κενής στοίβας, push, pop- όταν υλοποιούνται με πίνακες και λίστες και:

(i) ο δείκτης $S.\texttt{top}$ δείχνει το δεύτερο από πάνω στοιχείο της στοίβας,

(ii) ο δείκτης $S.\texttt{top}$ δείχνει το τελευταίο στοιχείο της στοίβας.

Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 14. (i) Εξηγήστε λεπτομερώς την υλοποίηση με στοίβα του αλγορίθμου δυαδικής αναζήτησης στοιχείου σε ταξινομημένο πίνακα, για τις εισόδους:

- $A = [0, 2, 4, 6, 8, 10, 12, 14]$, $key = 3$,
- $B = [1, 2, 3, 4, 5, 6, 7, 8, 9]$, $key = 4$.

(ii) Σχεδιάστε αλγόριθμο που χρησιμοποιεί στοίβα, υλοποιημένη ως λίστα, για την δυαδική αναζήτηση.

Άσκηση 15. Η ακολουθία Lucas περιγράφεται από την αναδρομή:

$$L(n) := L(n-1) + L(n-2),\ L(0) = 2, L(1) = 1.$$

(i) Περιγράψτε αναδρομικό αλγόριθμο για την ακολουθία Lucas.

(ii) Τρέξτε τον αλγόριθμό σας, υλοποιπημένο με στοίβα, για $n = 4$.

(iii) Σχεδιάστε αλγόριθμο που υλοποιεί στοίβα για τον υπολογισμό του $L(n)$.

Άσκηση 16. Έστω ουρά υλοποιημένη με πίνακα.

- Περιγράψτε αλγόριθμο που σε κάθε enqueue διορθώνει τον πίνακα κατάλληλα, ώστε να μην προκύπτει ζήτημα με πίνακα που μοιάζει γεμάτος ενώ δεν είναι.
- Περιγράψτε αλγόριθμο που σε κάθε dequeue διορθώνει τον πίνακα κατάλληλα, ώστε να μην προκύπτει ζήτημα με πίνακα που μοιάζει γεμάτος ενώ δεν είναι.

Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 17. Έστω μια ουρά προτεραιότητας, κάθε κόμβος της οποίας περιέχει και μια τιμή προτεραιότητας p.

- Υλοποιείστε την enqueue ώστε η ουρά προτεραιότητας να χει τους κόμβους της πάντα σε αύξουσα σειρά βαθμού προτεραιότητας. Υλοποιείστε την dequeue εκμεταλλευόμενα αυτήν την ιδιότητα, ώστε να αφαιρείται ο κόμβος με τον μικρότερο βαθμό προτεραιότητας p.
- Υλοποιείστε την enqueue ώστε κάθε νέος κόμβος να μπαίνει τελευταίος στην ουρά. Υλοποιείστε την dequeue ώστε να αφαιρείται ο κόμβος με τον μικρότερο βαθμό προτεραιότητας p.

Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 18. Θεωρήστε την ακόλουθη αναπαράσταση μίας ουράς προτεραιότητας Q με το πολύ n-στοιχεία:

- Τα στοιχεία της ουράς αποθηκεύονται σε έναν δισδιάστατο πίνακα $2 \times n$, όπου η πρώτη γραμμή του περιέχει τα στοιχεία της ουράς και η δεύτερη τον βαθμό προτεραιότητάς τους. Στον πίνακα αυτόν αναφερόμαστε γράφοντας $Q.\texttt{matrix}$. Για παράδειγμα το στοιχείο $Q.matrix[1, i]$ είναι το i-οστό στοιχείο της ουράς και το στοιχείο $Q.matrix[2, i]$ είναι ο βαθμός προτεραιότητάς του.
- Πέρα από τον πίνακα χρησιμοποιούμε δύο ακέραιους δείκτες $Q.\texttt{front}$ και $Q.\texttt{rear}$ με τιμή την στήλη του πίνακα που βρίσκεται το πρώτο και το τελευταίο στοιχείο της ουράς αντίστοιχα.

Δώστε τους αλγορίθμους που υλοποιούν τις πράξεις εισαγωγής και εξαγωγής σύμφωνα με αυτή την αναπαράσταση.

Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

4 Δέντρα

Άσκηση 19. (i) Δείξτε ότι ένα σχεδόν πλήρες δυαδικό δέντρο με n κορυφές, όπου στο κάτω κάτω επίπεδο οι κορυφές βρίσκονται όσο πιο αριστερά γίνεται, έχει τουλάχιστον $n/2$ φύλλα.

(ii) Έστω δυαδικό δέντρο $T = (V, E)$ με ρίζα, όπου κάθε κορυφή έχει είτε 0 είτε 2 παιδιά. Έστω $f(T)$ το πλήθος των κορυφών του T με δύο παιδιά και $l(T)$ το πλήθος των φύλλων του. Δείξτε ότι $l(T) = f(T) + 1$.

Άσκηση 20. Να γραφούν αλγόριθμοι που, σε ένα δέντρο υλοποιημένο με συνδεδεμένη λίστα, υπολογίζει:

- το πλήθος των εσωτερικών κόμβων του δέντρου,
- το πλήθος των φύλλων του δέντρου,
- το ύψος του δέντρου.

Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 21. Σχεδιάστε αλγόριθμο που δέχεται σαν είσοδο ένα δυαδικό δέντρο (με στοιχεία φυσικούς αριθμούς) και ελέγχει αν είναι δυαδικό δέντρο αναζήτησης. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του αλγορίθμου σας.

Άσκηση 22. Σχεδιάστε αλγόριθμο που, με είσοδο ένα δέντρο αναζήτησης και τις διευθύνσεις δύο κορυφών του δέντρου, επιστρέφει το μικρότερο δυνατό υποδέντρο που περιέχει και τις δύο κορυφές.

Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του αλγορίθμου σας.

Άσκηση 23. Σχεδιάστε αλγόριθμο που, με είσοδο ένα δέντρο αναζήτησης και την διεύθυνση μιας κορυφής του δέντρου, βρίσκει την διεύθυνση της κορυφής εκείνης με την αμέσως μεγαλύτερη τιμήμ αν αυτή υπάρχει. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του αλγορίθμου σας.

Άσκηση 24. Σχεδιάστε αλγόριθμο που, με είσοδο ένα τριαδικό δέντρο, το μετατρέπει σε δυαδικό. Πόσους νέους κόμβους θα χρειαστεί να προσθέσετε; Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά του αλγορίθμου σας.

Άσκηση 25. Να υλοποιηθεί ουρά προτεραιότητας με σωρό και να περιγραφούν αλγόριθμοι για τις enqueue και dequeue. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 26. Να βρεθεί άνω και κάτω φράγμα για το ύψος ενός δέντρου T αν:

- T AVL,
- T κοκκινόμαυρο.

Άσκηση 27. Να υλοποιηθούν αλγόριθμοι για την εισαγωγή και εξαγωγή κόμβουν σε δέντρα AVL. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 28. Να υλοποιηθούν αλγόριθμοι για την εισαγωγή και εξαγωγή κόμβουν σε κοκκινόμαυρα δέντρα. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 29. Να γραφούν αλγόριθμοι που υλοποιούν την εύρεση αντιπροσώπου και την ένωση συνόλων, όταν τα ξένα σύνολα αναπαριστώνται με:

- συνδεδεμένη λίστα,
- δέντρα.

5 Αναπαράσταση γραφημάτων

Να γραφούν αλγόριθμοι για τις ακόλουθες πράξεις:

I) προσθήκη κορυφής,

II) αφαίρεση κορυφής,

III) προσθήκη ακμής,

IV) αφαίρεση ακμής,

V) έλεγχος αν δυο κορυφές είναι γειτονικές,

VI) εύρεση βαθμού κορυφής,

στην περίπτωση που το $G = (V, E)$ είναι:

i) μη-κατευθυνόμενο,

ii) κατευθυνόμενο

και που αναπαρίσταται με:

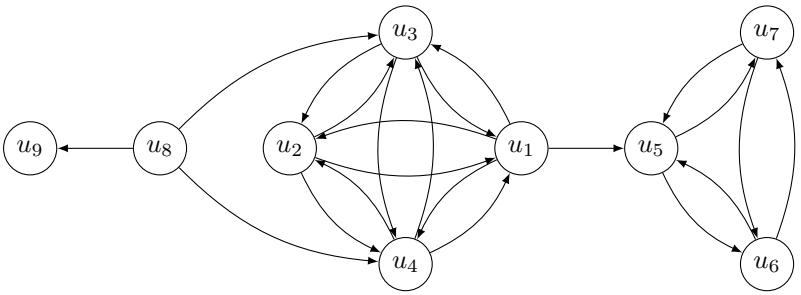
a) πίνακα γειτνίασης,

b) πίνακα πρόσπτωσης,

c) λίστα γειτνίασης.

Άσκηση 30. Αποδείξτε ότι για ένα κατευθυνόμενο γράφημα $D = (G, V)$ με πίνακα γειτνίασης A, το πλήθος των περιπάτων, δηλαδή των μονοπατιών που μπορούμε να επαναλάβουμε κορυφές, δίνεται από τον A^n.

Άσκηση 31. Εκτελέστε τους DFS και BFS για το γράφημα:



υλοποιώντας τους με στοίβα και ουρά αντίστοιχα.

Άσκηση 32. Να γραφεί αλγόριθμος που με είσοδο ένα κατευθυνόμενο γράφημα $D = (V, E)$, ελέγχει αν υπάρχουν κύκλοι σε αυτό. Μπορείτε να τον τροποποιήσετε ώστε να τους επιστρέφει; Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 33. Να γραφεί αλγόριθμος που με είσοδο ένα κατευθυνόμενο γράφημα $D = (V, E)$, να βρίσκει τις Ισχυρά Συνεκτικές του Συνιστώσες. Υπολογίστε και εξηγήστε την χρονική πολυπλοκότητά των αλγορίθμων σας.

Άσκηση 34. Να υλοποιηθεί ο αλγόριθμος **MaxDelete** όπου, με είσοδο ένα συνεκτικό βεβαρυμένο γράφημα, διαγράφει τις βαρύτερες ακμές σε φθίνουσα σειρά όσο δεν σπάει η συνεκτικότητα.

Άσκηση 35. Έστω συνεκτικό βεβαρυμένο γράφημα με βάρη και στις κορυφές και στις ακμές. Να υπολογιστεί ένα ελάχιστο επικαλύπτον δέντρο του γραφήματος, που ελαχιστοποιεί το συνολικό άθροισμα βαρών κορυφών και ακμών.