

ΠΡΟΣΟΜΟΙΩΣΗ 2025 ΣΧΟΛΕΙΟ

A1. ΣΛΣΣΛΛ A2. Δδγ

A3.

i. (04/25)

- Η τυχαία πρόσβαση στη λίστα δεν επιτρέπεται. Είναι αδύνατο να φτάσετε στον n -οστό κόμβο μιας απλά συνδεδεμένης λίστας χωρίς πρώτα να περάσετε από όλους τους κόμβους διαδοχικά μέχρι τον συγκεκριμένο κόμβο ξεκινώντας από τον πρώτο κόμβο. Εναλλακτικά, στην περίπτωση της διπλά συνδεδεμένης λίστας μπορείτε να ξεκινήσετε και από τον τελευταίο κόμβο. Επομένως, δεν μπορούμε να πραγματοποιήσουμε με αποτελεσματικό τρόπο δυαδική αναζήτηση σε συνδεδεμένες λίστες.
- Οι συνδεδεμένες λίστες έχουν πολύ μεγαλύτερη επιβάρυνση από τους πίνακες, αφού οι συνδεδεμένοι κόμβοι της λίστας είναι δυναμικά και κάθε κόμβος στη λίστα πρέπει, επιπλέον, να αποθηκεύσει έναν πρόσθετο δείκτη που θα δείχνει στον επόμενο κόμβο. Στην περίπτωση των διπλά συνδεδεμένων λιστών χρειαζόμαστε επιπλέον έναν δεύτερο δείκτη που θα δείχνει στον προηγούμενο κόμβο.

ii. (04/25)

- Κάθε πρόγραμμα που γράφεται σε οποιαδήποτε γλώσσα προγραμματισμού, πρέπει να μετατραπεί σε μορφή αναγνωρίσιμη και εκτελέσιμη από τον υπολογιστή, δηλαδή σε ακολουθίες δυαδικών ψηφίων 0 και 1 (τις εντολές **γλώσσας μηχανής**)
- Σε περίπτωση που ο μεταγλωττιστής ή ο διερμηνευτής συναντήσει συντακτικά λάθη στο πηγαίο πρόγραμμα, σταματά τη μεταγλώττιση και εμφανίζει κατάλληλα διαγνωστικά μηνύματα.

A4. (05/25)

- είναι δυαδικό δένδρο διότι είναι διατεταγμένο και κάθε κόμβος του έχει το πολύ 2 παιδιά
- δεν είναι δυαδικό δένδρο αναζήτησης διότι 'M' < 'N'

B1 (04/25) (04/25)

ΔΙΑΒΑΣΕ X

$Y \leftarrow 10$

ΑΝ $Y \geq X$ ΤΟΤΕ

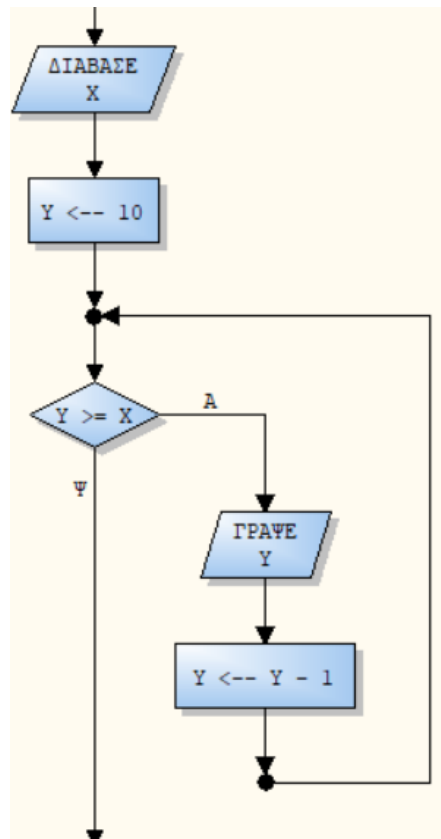
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ Y

$Y \leftarrow Y - 1$

ΜΕΧΡΙΣ_ΟΤΟΥ $Y < X$

ΤΕΛΟΣ_ΑΝ



B2 α. OS, χωρητικότητα, β. βοήθεια(), συνομιλία() γ. Λ, δ. Λ, ε. Λ

B4

!
!
D
O
O
G

top=6

ΠΡΟΓΡΑΜΜΑ θεμαΓ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: αποδ, απορ

ΠΡΑΓΜΑΤΙΚΕΣ: β1, β2, β3, βσ, max,

ποσ

ΑΡΧΗ

αποδ <- 0

απορ <- 0

max <- 0

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ β1, β2, β3

ΑΝ ok(β1, β2, β3) = ΑΛΗΘΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'αποδεκτή'

αποδ <- αποδ + 1

βσ <- β1 + β2 + β3

ΑΝ βσ > max ΤΟΤΕ

max <- βσ

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'απορρίπτεται'

απορ <- απορ + 1

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ αποδ = 100

ποσ <- απορ / (απορ + αποδ) * 100

ΓΡΑΨΕ ποσ, max

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ ok(α, β, γ): ΛΟΓΙΚΗ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: π

ΠΡΑΓΜΑΤΙΚΕΣ: α, β, γ

ΑΡΧΗ

π <- 0

ok <- ΑΛΗΘΗΣ

ΑΝ α < 150 Η α > 200 ΤΟΤΕ

π <- π + 1

ΤΕΛΟΣ_ΑΝ

ΑΝ β < 150 Η β > 200 ΤΟΤΕ

π <- π + 1

ΤΕΛΟΣ_ΑΝ

ΑΝ γ < 150 Η γ > 200 ΤΟΤΕ

π <- π + 1

ΤΕΛΟΣ_ΑΝ

ΑΝ π >= 2 ΤΟΤΕ

ok <- ΨΕΥΔΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΠΡΟΓΡΑΜΜΑ θεμαΔ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j, last, ep[10], pl[10], temp

ΠΡΑΓΜΑΤΙΚΕΣ: λ[8, 10], σ, max

ΧΑΡΑΚΤΗΡΕΣ: ON[8], maxON[10], temp2

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8

ΔΙΑΒΑΣΕ ON[i]

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ λ[i, j]

ΜΕΧΡΙΣ_ΟΤΟΥ λ[i, j] >= 0 ΚΑΙ λ[i, j] <= 63

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8

σ <- 0

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10

σ <- σ + λ[i, j]

ΑΝ λ[i, j] <> 0 ΤΟΤΕ

last <- j

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ ON[i], σ, last

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10

ep[j] <- j

!αριθμοί επεισοδίων

pl[j] <- 0

!πλήθος πρωταγωνιστών

max <- 0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8

ΑΝ λ[i, j] <> 0 ΤΟΤΕ

pl[j] <- pl[j] + 1

ΤΕΛΟΣ_ΑΝ

ΑΝ λ[i, j] > max ΤΟΤΕ

max <- λ[i, j]

maxON[j] <- ON[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 10

ΓΙΑ j ΑΠΟ 10 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1

ΑΝ pl[j - 1] < pl[j] ΤΟΤΕ

temp <- pl[j - 1]

pl[j - 1] <- pl[j]

pl[j] <- temp

temp2 <- ON[j - 1]

ON[j - 1] <- ON[j]

ON[j] <- temp2

temp <- ep[j - 1]

ep[j - 1] <- ep[j]

ep[j] <- temp

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ