

2020

Νέα Ύλη

Algorithms Computer  
Science Laboratory -  
Theodoropoulos D.

Αλγόριθμος



## CORONAVIRUS PROGRAMMING

ΠΡΟΒΛΗΜΑ-ΣΠΟΥΔΑΙΟΤΗΤΑ ΑΛΓΟΡΙΘΜΩΝ- ΤΕΧΝΙΚΕΣ  
ΣΧΕΔΙΑΣΗΣ ΑΛΓΟΡΙΘΜΩΝ- ΣΤΟΙΒΑ-ΟΥΡΑ-ΕΚΣΦΑΛΜΑΤΩΣΗ

**[ ΠΡΟΣΑΡΜΟΓΗ ΤΗΣ ΥΛΗΣ- CORONAVIRUS ]**



**1. Να δώσετε τον ορισμό του προβλήματος.**

Πρόβλημα είναι μια κατάσταση που χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση της δεν είναι γνωστή ούτε προφανής.

**2. Τι ονομάζεται επίλυση ενός προβλήματος;**

Επίλυση (ή αντιμετώπιση) ενός προβλήματος ονομάζουμε τη διαδικασία μέσω της οποίας ο λύτης του προβλήματος βρίσκει το ζητούμενο.

**3. Η κατανόηση ενός προβλήματος αποτελεί συνάρτηση δύο παραγόντων. Να τους αναφέρετε; (Εξετάσεις 2009 & 2016)**

Είναι η σωστή διατύπωση εκ μέρους του δημιουργού του και η σωστή ερμηνεία εκ μέρους του λύτη. Διαφορετικά κάθε προσπάθεια επίλυσης του προβλήματος θα οδηγηθεί σε αποτυχία.

**4. Ποιο μέσο χρησιμοποιείται για τη διατύπωση ενός προβλήματος;**

Συνηθέστερα μέσα είναι ο λόγος είτε προφορικός είτε γραπτός.

**5. Τι εννοούμε με τον όρο «σαφήνεια διατύπωσης προβλήματος»;**

Ο λόγος ως μέσο επικοινωνίας και συνεννόησης πρέπει να χαρακτηρίζεται από σαφήνεια. Άστοχη χρήση ορολογίας και λανθασμένη σύνταξη είναι δύο στοιχεία που μπορούν να προκαλέσουν παρερμηνείες και παραπλανήσεις.

Σημαντικός ακόμα παράγοντας στη σωστή αντιμετώπιση ενός προβλήματος είναι η αποσαφήνιση του χώρου στον οποίο αναφέρεται.

Για παράδειγμα: «Ο Γιάννης και η Μαρία είναι παντρεμένοι». Πρώτη ερμηνεία: Ο Γιάννης και η Μαρία είναι παντρεμένοι μεταξύ τους. Δεύτερη ερμηνεία: Ο Γιάννης είναι παντρεμένος και η Μαρία είναι παντρεμένη.

Για τη σωστή **ερμηνεία**, σημαντικό ρόλο παίζει το επίπεδο γνώσεων και αντίληψης του λύτη, οι γενικές και ειδικές γνώσεις του, το ενδιαφέρον και η ευφυΐα του.  
 Η αποσαφήνιση του **χώρου** σχετίζεται με το αν ένα πρόβλημα είναι επιστημονικό ή της καθημερινής ζωής. Επιπλέον, τα δεδομένα του προβλήματος είναι αυτά που μας παρέχουν αυτή τη πληροφορία.

**6. Να δώσετε τους ορισμούς των παρακάτω όρων: (Εξετάσεις 2007)**

**α. Δεδομένο, β. Πληροφορία, γ. Επεξεργασία δεδομένων**

**α. Δεδομένο** είναι οποιοδήποτε στοιχείο μπορεί να γίνει αντιληπτό από έναν τουλάχιστον παρατηρητή με μία από τις πέντε αισθήσεις του.

**β. Πληροφορία** είναι οποιοδήποτε γνωσιακό στοιχείο προέρχεται από επεξεργασία δεδομένων.

**γ. Επεξεργασία** είναι η διαδικασία κατά την οποία ένας «μηχανισμός» δέχεται δεδομένα, τα επεξεργάζεται και αποδίδει πληροφορίες (είναι δηλαδή η συστηματική εκτέλεση πράξεων στα δεδομένα). Για παράδειγμα, ο ανθρώπινος εγκέφαλος ή ο υπολογιστής.



## 7. Να δώσετε τον ορισμό της δομής ενός προβλήματος. (Εξετάσεις 2014)

Δομή προβλήματος είναι τα συστατικά του μέρη, τα επιμέρους τμήματα που το αποτελούν, καθώς και ο τρόπος με τον οποίο συνδέονται μεταξύ τους.

## 8. Πως βοηθά στην επίλυση ενός προβλήματος η καταγραφή της δομής του;

Η κατανόηση ενός προβλήματος είναι βασική προϋπόθεση για να γίνει στη συνέχεια δυνατή η σωστή αποτύπωση της δομής του. Η καταγραφή της δομής ενός προβλήματος σημαίνει αυτόματα ότι έχει αρχίσει η διαδικασία ανάλυσης του προβλήματος σε άλλα απλούστερα. Με τη σειρά τους τα νέα προβλήματα μπορούν να αναλυθούν σε άλλα, ακόμη πιο απλά. Η δυσκολία αντιμετώπισης των προβλημάτων ελαττώνεται όσο περισσότερο προχωράει η ανάλυση τους σε απλούστερα προβλήματα. Η διαδικασία αυτή μπορεί να συνεχιστεί μέχρις ότου τα επιμέρους προβλήματα που προέκυψαν θεωρηθούν αρκετά απλά και η αντιμετώπισή τους χαρακτηρίζεται ως δυνατή.

## 9. Να αναφέρετε τους τρόπους περιγραφής και αναπαράστασης των προβλημάτων. Να δώσετε ένα παράδειγμα.

- **Φραστικά**, δηλαδή στη φυσική γλώσσα που μιλάμε.
- **Διαγραμματικά**, δηλαδή ένα διάγραμμα σε σχήμα γενεαλογικού δέντρου στο οποίο:
  - ✓ Το αρχικό πρόβλημα αναπαρίσταται από ένα ορθογώνιο παραλληλόγραμμο.
  - ✓ Κάθε ένα από τα υποπροβλήματα, στα οποία διαχωρίζεται το πρόβλημα αναπαρίσταται επίσης από ένα ορθογώνιο παραλληλόγραμμο.
  - ✓ Τα παραλληλόγραμμο που αντιστοιχούν στα απλούστερα προβλήματα, στα οποία αναλύεται ένα πρόβλημα, τίθενται ένα επίπεδο χαμηλότερα.

Γιατί είναι χρήσιμη η διαγραμματική αναπαράσταση;

Η διαγραμματική αναπαράσταση βοηθά στην καλύτερη κατανόηση του προβλήματος, καθώς και στη σχεδίαση της λύσης του, αφού προσφέρεται με τον τρόπο αυτό μια απτή απεικόνιση της **δομής** του προβλήματος.

## 10. Να αναλύσετε φραστικά και διαγραμματικά το πρόβλημα «Αντιμετώπιση ναρκωτικών».

### Φραστική αναπαράσταση

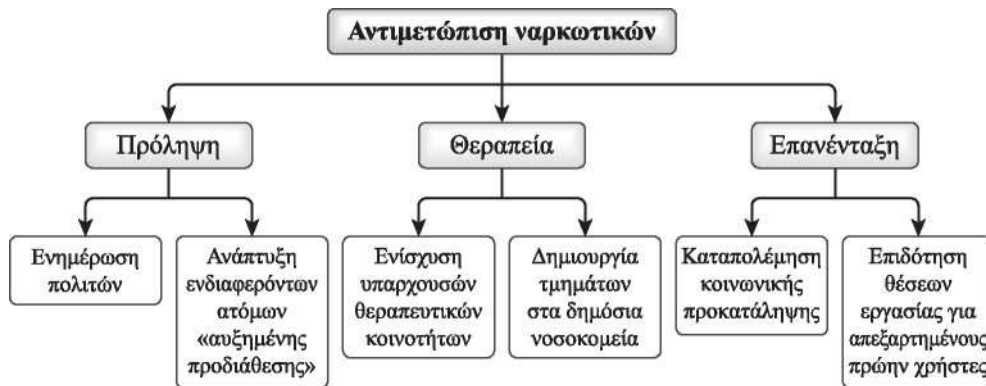
Το κύριο πρόβλημα είναι η αντιμετώπιση των ναρκωτικών. Αυτό το πρόβλημα μπορεί να γίνει απλούστερο αν διασπαστεί σε τρία υποπροβλήματα,

1. Πρόληψη
2. Θεραπεία
3. Επανένταξη

Τα τρία αυτά υποπροβλήματα μπορούν να διασπαστούν ως εξής:

- 1.1 Ενημέρωση πολιτών
- 1.2 Προσοχή στα άτομα αυξημένης προδιάθεσης
- 2.1 Ενίσχυση υπαρχουσών θεραπευτικών κοινοτήτων
- 2.2 Δημιουργία τμημάτων στα δημόσια νοσοκομεία
- 3.1 καταπολέμηση κοινωνικής προκατάληψης
- 3.2 Επιδότηση θέσεων εργασίας για πρώην χρήστες

Διαγραμματική αναπαράσταση



**11. Τι εννοούμε με τον όρο «καθορισμός απαιτήσεων προβλήματος»;**

Η σωστή επίλυση ενός προβλήματος προϋποθέτει τον επακριβή προσδιορισμό των δεδομένων τα οποία παρέχει το πρόβλημα και τη λεπτομερειακή καταγραφή των ζητούμενων που αναμένονται ως αποτελέσματα της επίλυσης του προβλήματος.

Η αποσαφήνιση των δεδομένων και των ζητούμενων του προβλήματος συχνά απαιτεί μια σειρά από **διευκρινιστικές ερωτήσεις** εκ μέρους του λύτη, είτε προς τον δημιουργό του προβλήματος είτε προς τον ίδιο του τον εαυτό, αφού πολλές φορές τα δεδομένα πρέπει να "ανακαλυφθούν" στα λεγόμενα του προβλήματος. Η διαδικασία αυτή απαιτεί προσοχή και συγκέντρωση στην σκέψη. Μεθοδολογία προσδιορισμού των δεδομένων ενός προβλήματος **δεν υπάρχει**, ούτε και μεθοδολογία εντοπισμού και αποσαφήνισης των ζητούμενων του προβλήματος.

**12. Να περιγράψετε τα στάδια αντιμετώπισης ενός προβλήματος**

- **Κατανόηση:** Σωστή και πλήρης αποσαφήνιση των δεδομένων και των ζητούμενων του προβλήματος.
- **Ανάλυση:** Το αρχικό πρόβλημα διασπάται σε επιμέρους απλούστερα προβλήματα, μέχρις ότου να είναι εύκολο να επιλυθούν.
- **Επίλυση:** Υλοποιείται η λύση του προβλήματος μέσω της λύσης των επιμέρους προβλημάτων (που έχουν διαχωριστεί στο στάδιο της ανάλυσης).



Στην καθημερινή ζωή και σε όλες τις επιστήμες συναντάμε την έννοια του προβλήματος. Οι απαιτούμενες ενέργειες για την αντιμετώπιση ενός προβλήματος φαίνονται στο παρακάτω σχήμα. Στην **είσοδο** εισάγονται τα δεδομένα του προβλήματος τα οποία στη συνέχεια θα υποστούν επεξεργασία. Στον **έλεγχο**, τα δεδομένα ελέγχονται ως προς την ορθότητά τους και γίνονται οι απαραίτητες διορθώσεις αν απαιτείται. Κατά την **επεξεργασία**, πραγματοποιούνται οι απαραίτητοι υπολογισμοί για να προκύψουν τα ζητούμενα αποτελέσματα και στην **έξοδο** πραγματοποιείται η εξαγωγή των αποτελεσμάτων.



### 13. Γιατί είναι σημαντική η έννοια του αλγορίθμου για την πληροφορική;

Η έννοια του αλγορίθμου είναι θεμελιώδης για την επιστήμη της πληροφορικής, αφού η δημιουργία προγραμμάτων αποτελεί μεγάλο μέρος της επιστήμης αυτής και ο αλγόριθμος αποτελεί τον θεμέλιο λίθο κάθε προγράμματος. Η μελέτη λοιπόν των αλγορίθμων αποτελεί κύριο μέλημα των επιστημόνων της πληροφορικής.

### 14. Από ποιες σκοπιές μελετά τους αλγορίθμους η Πληροφορική;

α. **Υλικού:** Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες υλικού και την αρχιτεκτονική του υπολογιστή, (όπως το αν ο υπολογιστής έχει κρυφή μνήμη και πόση ή τη ταχύτητα της κύριας και δευτερεύουσας μνήμης κλπ.)

β. **Γλωσσών προγραμματισμού:** Το είδος της γλώσσας προγραμματισμού που χρησιμοποιείται (δηλαδή χαμηλότερου ή υψηλότερου επιπέδου) αλλάζει τη δομή και τον αριθμό των εντολών ενός αλγορίθμου. Γενικά μια γλώσσα χαμηλότερου επιπέδου (όπως assembly) είναι ταχύτερη από μια άλλη γλώσσα που είναι υψηλότερου επιπέδου (όπως η Basic ή η Pascal).

γ. **Θεωρητική:** Διερευνά αν υπάρχει ή όχι αποδοτικός αλγόριθμος για την επίλυση ενός προ-βλήματος. Η προσέγγιση προσδιορίζει τα όρια της λύσης που θα βρεθεί σε συγκεκριμένο πρόβλημα.

δ. **Αναλυτική:** Μελετώνται οι υπολογιστικοί πόροι που απαιτούνται από έναν αλγόριθμο (π.χ. το μέγεθος κύριας και δευτερεύουσας μνήμης, ο χρόνος λειτουργίας της CPU και των λειτουργιών εισόδου / εξόδου κ.λπ.).

### 15. Τι είναι προγραμματισμός τι είναι πρόγραμμα;

Ο προγραμματισμός ασχολείται με τη δημιουργία του προγράμματος, δηλαδή του συνόλου των εντολών που πρέπει να δοθούν στον υπολογιστή για την επίλυση ενός προβλήματος. Το πρόγραμμα το οποίο γράφεται σε κάποια γλώσσα προγραμματισμού, δεν είναι απλώς η υλοποίηση του αλγορίθμου, αλλά βασικά στοιχεία του είναι τα δεδομένα και οι δομές δεδομένων επί των οποίων ενεργεί.

### 16. Ποιά στάδια περιλαμβάνει η επίλυση ενός προβλήματος με τον υπολογιστή;

- Περιλαμβάνει τα εξής στάδια: Προσδιορισμό του προβλήματος
- Ανάπτυξη του αντίστοιχου αλγορίθμου
- Συγγραφή προγράμματος (μορφή κατανοητή από τον υπολογιστή).

### 17. Ποιές είναι οι στοιχειώδεις ενέργειες που μπορεί να κάνει ένας υπολογιστής;

Ο υπολογιστής είναι μία μηχανή που καταλαβαίνει μόνο δύο καταστάσεις, οι οποίες αντιπροσωπεύονται με δύο αριθμούς το μηδέν και το ένα, τα ψηφία του δυαδικού συστήματος. Το μόνο πράγμα που κάνει ο υπολογιστής είναι στοιχειώδεις ενέργειες σε ακολουθίες αυτών των δύο ψηφίων, αλλά αυτές τις ενέργειες τις εκτελεί με ασύλληπτη ταχύτητα. Ο υπολογιστής μπορεί απλά να αποθηκεύει στη μνήμη τις ακολουθίες των δυαδικών ψηφίων, να τις ανακτά, να κάνει στοιχειώδεις αριθμητικές πράξεις με αυτές και να τις συγκρίνει.

**18. Τι είναι δεδομένα και τι πληροφορία; Ποιος ο ρόλος του αλγορίθμου στην παραγωγή πληροφορίας;**

**Δεδομένα** είναι ακατέργαστα γεγονότα, αφαιρετική αναπαράσταση (απλοποιημένη όψη) της πραγματικότητας, των οποίων κάθε φορά η επιλογή εξαρτάται από τον τύπο του προβλήματος. Οι **πληροφορίες** παράγονται από τα δεδομένα και είναι αποτέλεσμα επεξεργασίας και συσχετισμού των δεδομένων. Από πληροφορίες μπορούν να παραχθούν νέες πληροφορίες. Ο αλγόριθμος αποτελεί το μέσο για τη παραγωγή πληροφορίας.

**19. Να αναφέρετε τις σκοπιές από τις οποίες μελετά τα δεδομένα η πληροφορική;**

**α. Υλικού:** Το υλικό επιτρέπει στα δεδομένα ενός προγράμματος να αποθηκεύονται στην κύρια μνήμη και στις περιφερειακές συσκευές του υπολογιστή με διάφορες αναπαραστάσεις (π.χ. κώδικας ASCII).

**β. Γλωσσών προγραμματισμού:** Οι γλώσσες προγραμματισμού υψηλού επιπέδου επιτρέπουν τη χρήση διαφόρων τύπων μεταβλητών για να περιγράψουν ένα δεδομένο. Ο μεταφραστής κάθε γλώσσας είναι υπεύθυνος για την αποδοτικότερη μορφή αποθήκευσης κάθε μεταβλητής στον υπολογιστή.

**γ. Δομών δεδομένων:** Δομή δεδομένων είναι ένα σύνολο δεδομένων μαζί με ένα σύνολο επιτρεπτών λειτουργιών επί αυτών. Για παράδειγμα, μια τέτοια δομή είναι η εγγραφή που μπορεί να περιγράψει ένα είδος, πρόσωπο κλπ. Η εγγραφή αποτελείται από πεδία που αποθηκεύουν χαρακτηριστικά διαφορετικού τύπου (π.χ. κωδικός, περιγραφή κλπ.). Άλλη μορφή δομής δεδομένων είναι το αρχείο που αποτελείται από ένα σύνολο εγγραφών.

**δ. Ανάλυση δεδομένων:** Μελετώνται τρόποι καταγραφής και αλληλοσυσχέτισης των δεδομένων, έτσι ώστε να αναπαρασταθεί η γνώση για πραγματικά γεγονότα. Οι τεχνολογίες των *Βάσεων Δεδομένων*, της *Μοντελοποίησης Δεδομένων* και της *Αναπαράστασης Γνώσης* ανήκουν σε αυτή τη σκοπιά μελέτης των δεδομένων.

**20. Τι μελετά ο κλάδος της θεωρίας πληροφοριών;**

Η θεωρία πληροφοριών (Information theory) είναι ένα ιδιαίτερα σημαντικό πεδίο της πληροφορικής. Αντικείμενο μελέτης της αποτελούν η μέτρηση, η κωδικοποίηση και η μετάδοση της πληροφορίας.

**21. Να περιγραφεί η εντολή επίλεξε.**

Επίλεξε έκφραση	<Έκφραση>
περίπτωση λίστα τιμών 1 ομάδα εντολών 1	Είναι συνήθως μια μεταβλητή οποιουδήποτε τύπου, Στη γενική όμως περίπτωση μπορεί να είναι: <ul style="list-style-type: none"> <li>• Αριθμητική πράξη (π.χ. <b>επίλεξε</b> <math>x/T\_P(25)</math>)</li> <li>• Συγκριτική πράξη (π.χ. <b>επίλεξε</b> <math>x&gt;0</math>).</li> <li>•</li> </ul>
περίπτωση λίστα τιμών 2 ομάδα εντολών 2 .....	
περίπτωση λίστα τιμών N ομάδα εντολών N	Η έκφραση που βρίσκεται δίπλα στη δεσμευμένη λέξη <b>ΕΠΙΛΕΞΕ</b> μπορεί να είναι μεταβλητή ( <b>ΕΠΙΛΕΞΕ</b> $x$ ), αριθμητική πράξη ( <b>ΕΠΙΛΕΞΕ</b> $x \bmod 2$ ) ή συγκριτική πράξη ( <b>ΕΠΙΛΕΞΕ</b> $x < y$ ).
περίπτωση αλλιώς ομάδα εντολών N+1	
τέλος_επιλογών	



<Λίστα τιμών N>

Μπορεί να είναι:

- Συγκεκριμένες διακριτές τιμές:  
    **περίπτωση 1,2,5** (περιλαμβάνει τις τιμές 1,2 και 5)
- Διαστήματα τιμών **από** τιμή 1 **έως** τιμή 2:  
    **περίπτωση από 1 έως 20** (περιλαμβάνει όλες τις τιμές [1,20])
- Διαστήματα τιμών τιμή1..τιμή2:  
    **περίπτωση 1..20** (περιλαμβάνει όλες τις τιμές [1,20])
- Τιμές που ικανοποιούν μια συνθήκη:  
    **περίπτωση > 0** (περιλαμβάνει όλες τις θετικές τιμές)

Η λειτουργία της επίλεξε είναι η εξής:

- Αρχικά υπολογίζεται η τιμή μιας έκφρασης.
- Στη συνέχεια ανάλογα σε ποιά λίστα τιμών (λίστα τιμών 1 έως λίστα τιμών N) βρίσκεται η τιμή αυτή, εκτελείται η αντίστοιχη ομάδα εντολών.
- Σε περίπτωση που η τιμή της έκφρασης δεν ανήκει σε καμία λίστα τιμών, τότε εκτελούνται οι εντολές που βρίσκονται κάτω από την περίπτωση *αλλιώς*, η οποία όμως είναι προαιρετική. Στην περίπτωση που δεν υπάρχει αυτή η επιλογή, τότε δεν εκτελείται καμία εντολή.
- Σε όλες τις περιπτώσεις η εκτέλεση συνεχίζεται με τις εντολές που βρίσκονται μετά το τέλος επιλογών.

Η χρήση της **ΕΠΙΛΕΞΕ** λόγω τις συμπαγούς δομής της προσφέρει σημαντικά πλεονεκτήματα στον προγραμματισμό και χρησιμοποιείται κυρίως όταν μια έκφραση παίρνει πολλές διακριτές τιμές.  
Εάν έχουμε σύνθετες λογικές εκφράσεις χρησιμοποιούμε την εντολή **ΑΛΛΙΩΣ\_ΑΝ** καθώς η **ΕΠΙΛΕΞΕ** δεν μπορεί να τις χειριστεί.

## 22. Να ορίσετε την δομή δεδομένων στοίβα. Ποιά λειτουργία επιτελεί η μέθοδος επεξεργασίας LIFO.

**Στοιβα** (stack) ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που βρίσκονται στη κορυφή της στοίβας να λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας να λαμβάνονται τελευταία, όπως μια στοίβα πιάτων. Η στοίβα χρησιμοποιεί τη μέθοδο επεξεργασίας "τελευταίο μέσα, πρώτο έξω" LIFO (Last In, First Out), δηλαδή το στοιχείο που εισέρχεται τελευταίο εξέρχεται πρώτο.

Για την υλοποίηση της στοίβας με μονοδιάστατο πίνακα χρησιμοποιείται ένας δείκτης που δείχνει την θέση του τελευταίου στοιχείου που έχει εισαχθεί (κορυφή -top) αλλά και τη θέση του πρώτου στοιχείου προς απώθηση.

## 23. Να περιγράψετε τις κύριες λειτουργίες μιας στοίβας.

Οι κύριες λειτουργίες είναι η **ώθηση** στοιχείου στην κορυφή της στοίβας και η **απώθηση** στοιχείου από την κορυφή της στοίβας. Κατά την ώθηση πρέπει να ελέγχεται αν υπάρχει χώρος για εισαγωγή νέου στοιχείου (έλεγχος υπερχειλίσης) και κατά την απώθηση αν υπάρχει τουλάχιστον ένα τουλάχιστον στοιχείο για διαγραφή (έλεγχος υποχείλισης). Ο λόγος εμφάνισης υπερχειλίσης ή υποχείλισης είναι ότι για την υλοποίηση της δομής της στοίβας στη ΓΛΩΣΣΑ χρησιμοποιείται μονοδιάστατος πίνακας, που είναι στατική δομή και έχει σταθερό μέγεθος.

## 24. Να ορίσετε την δομή δεδομένων ουρά. Ποιά λειτουργία επιτελεί η μέθοδος επεξεργασίας FIFO.

**Ουρά** (queue) ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα. Η ουρά λειτουργεί όπως μια ουρά ανθρώπων σε μια τράπεζα. Χρησιμοποιεί την τεχνική "πρώτο μέσα, πρώτο έξω" FIFO (First In, First Out), δηλαδή το στοιχείο που εισέρχεται πρώτο εξέρχεται πρώτο.

Για την υλοποίηση της ουράς με μονοδιάστατο πίνακα χρησιμοποιούνται δυο δείκτες ο εμπρός (front) και ο πίσω (rear) που δείχνουν αντίστοιχα την θέση του πρώτου στοιχείου που θα εξαχθεί και την θέση του τελευταίου στοιχείου που έχει εισαχθεί.

## 25. Να περιγράψετε τις κύριες λειτουργίες μιας ουράς.

Οι κύριες λειτουργίες είναι η **εισαγωγή** στοιχείου στο πίσω άκρο της ουράς και η **εξαγωγή** στοιχείου από το μπροστινό άκρο της ουράς. Για την υλοποίηση της δομής της ουράς στη ΓΛΩΣΣΑ χρησιμοποιείται μονοδιάστατος πίνακας, που είναι στατική δομή και έχει σταθερό μέγεθος. Κατά την εισαγωγή πρέπει να ελέγχεται αν υπάρχει ελεύθερος χώρος στον πίνακα (έλεγχος για το εάν η ουρά είναι γεμάτη) και κατά την εξαγωγή αν υπάρχει τουλάχιστον ένα τουλάχιστον στοιχείο στην ουρά (έλεγχος για το εάν η ουρά είναι άδεια).

## 26. Γιατί απαιτείται η ανάλυση ενός προβλήματος;

Ένα πρόβλημα επιλύεται με περισσότερες από μια λύσεις και κάθε λύση μπορεί να προέλθει με ποικίλες και διαφορετικές προσεγγίσεις, τεχνικές και μεθόδους. Η ανάλυση κάθε προβλήματος είναι απαραίτητη, για να αναζητηθεί η πιο κατάλληλη μέθοδος που θα παρέχει τη ζητούμενη λύση, όσο γίνεται ταχύτερα και με το λιγότερο δυνατό κόστος σε υπολογιστικούς πόρους.

## 27. Ποιά είναι τα πλεονεκτήματα των μεθόδων ανάλυσης προβλημάτων;

Εντοπίζεται η πλέον κατάλληλη και αποδοτική λύση και έτσι έχουμε την εξοικονόμηση υπολογιστικών πόρων και κόστους για την επίλυση.

## 28. Τι περιλαμβάνει η ανάλυση ενός προβλήματος σε ένα σύγχρονο προγραμματιστικό περιβάλλον;

Η ανάλυση ενός προβλήματος σε ένα σύγχρονο προγραμματιστικό περιβάλλον περιλαμβάνει:

1. την καταγραφή της υπάρχουσας πληροφορίας για το πρόβλημα,
2. την αναγνώριση των ιδιοτήτων του προβλήματος,
3. την αποτύπωση των συνθηκών και προϋποθέσεων υλοποίησής του προβλήματος,
4. την πρόταση επίλυσης με τη χρήση κάποιας μεθόδου,
5. και την τελική επίλυση με χρήση υπολογιστικών συστημάτων.

## 29. Υπάρχει ένας ενιαίος κανόνας που να αναφέρεται στην επίλυση όλων των προβλημάτων; Ποιά προβλήματα ονομάζονται συγγενή;

Δεν υπάρχει ένας ενιαίος κανόνας, δηλαδή μια γενική φόρμουλα που να αναφέρεται στην επίλυση του συνόλου των προβλημάτων. Υπάρχουν απλώς τα συγγενή



προβλήματα, δηλαδή αυτά που μπορούν να αναλυθούν με παρόμοιο τρόπο και να αντιμετωπιστούν με αντίστοιχες προσεγγίσεις, τεχνικές και μεθόδους.

### 30. Κατά την ανάλυση του προβλήματος σε ποιές ερωτήσεις πρέπει να δοθούν απαντήσεις;

Κατά την ανάλυση ενός προβλήματος πρέπει να δοθούν απαντήσεις σε κάθε μια από τις παρακάτω ερωτήσεις:

1. Ποιά είναι τα δεδομένα και ποιό το μέγεθος του προβλήματος;
2. Ποιές είναι οι συνθήκες που πρέπει να πληρούνται για την επίλυση του προβλήματος;
3. Ποιά είναι η πλέον αποδοτική μέθοδος επίλυσης του προβλήματος (σχεδίαση αλγορίθμου);
4. Πώς θα καταγραφεί η λύση του προβλήματος (π.χ. σε ψευδογλώσσα);
5. Ποιός είναι ο τρόπος υλοποίησης της λύσης στο συγκεκριμένο υπολογιστικό σύστημα (π.χ. επιλογή γλώσσας προγραμματισμού);

### 31. Γιατί παρουσιάζουν ιδιαίτερο ενδιαφέρον οι μέθοδοι ανάλυσης και επίλυσης προβλημάτων;

Παρουσιάζουν ιδιαίτερο ενδιαφέρον για τους εξής λόγους:

1. Παρέχουν ένα γενικό πρότυπο το οποίο είναι κατάλληλο για την επίλυση προβλημάτων ευρείας κλίμακας,
2. μπορούν να αναπαρασταθούν με κοινές δομές δεδομένων και ελέγχου οι οποίες υποστηρίζονται από τις σύγχρονες γλώσσες προγραμματισμού,
3. παρέχουν τη δυνατότητα καταγραφής των χρονικών και χωρικών απαιτήσεων της μεθόδου επίλυσης, για να μπορεί να γίνει επακριβής εκτίμηση των αποτελεσμάτων.

### 32. Τι είναι η μέθοδος *Διαίρει και Βασίλευε*;

Η **Διαίρει και Βασίλευε** (divide and conquer) αποτελεί μια μέθοδο σχεδίασης αλγορίθμων στην οποία εντάσσονται οι τεχνικές που υποδιαιρούν ένα πρόβλημα σε μικρότερα υποπροβλήματα, που έχουν την ίδια τυποποίηση με το αρχικό πρόβλημα αλλά είναι μικρότερα σε μέγεθος. Με όμοιο τρόπο, τα υποπροβλήματα αυτά μπορούν να διαιρεθούν σε ακόμη μικρότερα υποπροβλήματα κοκ. Έτσι η επίλυση ενός προβλήματος έγκειται στη σταδιακή επίλυση των όσο το δυνατόν μικρότερων υποπροβλημάτων, ώστε τελικά να καταλήξουμε στη συνολική λύση του αρχικού ευρύτερου προβλήματος. Αυτή η προσέγγιση ονομάζεται από επάνω προς τα κάτω (top-down).

Πως υλοποιείται η μέθοδος **Διαίρει και Βασίλευε** στο πλαίσιο του μαθήματος;

Στο πλαίσιο του μαθήματος η υλοποίηση γίνεται με την επαναληπτική προσέγγιση δηλαδή με διαδοχικές επαναλήψεις.

### 33. Τι είναι η προσέγγιση της λύσης "από πάνω προς τα κάτω";

Η προσέγγιση της λύσης "από πάνω προς τα κάτω" είναι η τεχνική κατά την οποία κάθε πρόβλημα διαιρείται σε μικρότερα επιμέρους υποπροβλήματα και κάθε ένα από αυτά τα υποπροβλήματα διαιρείται σε ακόμα απλούστερα και μικρότερα υποπροβλήματα. Στο τέλος τα επιμέρους υποπροβλήματα είναι αρκετά απλά, ώστε η σταδιακή επίλυση τους, επιφέρει τη συνολική λύση του προβλήματος.

**34. Να αποδώσετε τα βήματα της εφαρμογής Διαίρει και Βασίλευε.**

Η περιγραφή αυτής της μεθόδου σχεδίασης αλγορίθμων μπορεί να αποδοθεί με τα επόμενα βήματα:

1. Δίνεται για επίλυση ένα στιγμιότυπο ενός προβλήματος.
2. Υποδιαιρείται το στιγμιότυπο του προβλήματος σε υπο-στιγμιότυπα του ίδιου προβλήματος.
3. Δίνεται ανεξάρτητη λύση σε κάθε ένα υπο-στιγμιότυπο.
4. Συνδυάζονται όλες οι μερικές λύσεις που βρέθηκαν για τα υπο-στιγμιότυπα, έτσι ώστε να δοθεί η συνολική λύση του προβλήματος.

**35. Που εφαρμόζεται κατά κύριο λόγο η μέθοδος Διαίρει και Βασίλευε;**

Η μέθοδος Διαίρει και Βασίλευε εφαρμόζεται κυρίως για τη γρήγορη αναζήτηση δεδομένων στην περίπτωση ενός ταξινομημένου συνόλου στοιχείων. Ένας κλασικός αλγόριθμος που ακολουθεί τη φιλοσοφία της είναι η **Δυναμική Αναζήτηση**.

**36. Πόσες συγκρίσεις (επαναλήψεις) απαιτούνται για την εύρεση ενός στοιχείου σε σύνολο n ταξινομημένων στοιχείων με τη μέθοδο Διαίρει και Βασίλευε;**

Ο μέγιστος αριθμός συγκρίσεων που απαιτούνται για την εύρεση ενός στοιχείου σε σύνολο n ταξινομημένων στοιχείων, συμπεριλαμβανομένης και της περίπτωσης μη ύπαρξης του στοιχείου, δίνεται από τον τύπο:

$$A_M(\log_2 n + 1)$$

Παραδείγματα:

Σύνολο στοιχείων	Μέγιστος αριθμός συγκρίσεων
100	$A_M(\log_2 100 + 1) = A_M(6.64 + 1) = A_M(7.64) = 7$
1000	$A_M(\log_2 1000 + 1) = A_M(9.96 + 1) = A_M(10.96) = 10$

Γενικά ισχύει,  $\log_2 8 = 3$ , εφόσον  $2^3 = 8$ . Με την ίδια λογική:

$$\log_2 2 = 1, \log_2 4 = 2, \log_2 8 = 3, \log_2 16 = 4, \log_2 32 = 5,$$

$$\log_2 64 = 6, \log_2 128 = 7, \log_2 256 = 8, \log_2 512 = 9, \log_2 1024 = 10$$

**37. Σε ποιές κατηγορίες διακρίνονται τα λάθη ενός προγράμματος;**

- Στα **συντακτικά** λάθη (λάθη κατά την συγγραφή, υλοποίηση του προγράμματος)
- Στα λάθη που οδηγούν σε **αντικανονικό τερματισμό** (λάθη κατά την εκτέλεση)
- Στα **λογικά** λάθη (λάθη που παράγουν λανθασμένα αποτελέσματα).

Η εμφάνιση των λαθών σε ένα πρόγραμμα, οφείλεται σε απρόβλεπτες καταστάσεις, σε κακούς χειρισμούς ή σε λογικά λάθη στον σχεδιασμό των προγραμμάτων.

Ένα πρόγραμμα πριν παραδοθεί για πραγματική λειτουργία, πρέπει να ελέγχεται και να είναι βεβαιωμένο ότι *δουλεύει απρόσκοπτα και παράγει σωστά αποτελέσματα*. Τα λάθη που παρουσιάζονται, μερικές φορές μπορεί να είναι ασήμαντα, μπορεί όμως να είναι και ιδιαίτερα κρίσιμα και να οδηγούν στη κατάρρευση του συστήματος.

### 38. Ποιά λάθη λέγονται συντακτικά και που οφείλονται;

Τα λάθη που προκύπτουν κατά τη συγγραφή του πηγαίου προγράμματος είναι τα συντακτικά λάθη, δηλαδή λάθη που παραβιάζουν τη σύνταξη εντολών, (για αυτό και ονομάζονται λάθη κατά την υλοποίηση).

#### Συντακτικά λάθη:

- Λανθασμένη γραφή μιας δεσμευμένης λέξης.
- Παράλειψη μιας δεσμευμένης λέξης.
- Λανθασμένη σύνταξη εκφράσεων (αριθμητικών-λογικών-αλφαριθμητικών).
- Παράλειψη δήλωσης μεταβλητών.
- Εκχώρηση λανθασμένου τύπου δεδομένων σε μια μεταβλητή.
- Λανθασμένος αριθμός παραμέτρων κατά την κλήση- δήλωση ενός υποπρογράμματος.
- Εκχώρηση τιμής σε συμβολική σταθερά.

### 39. Πώς ανιχνεύονται και αντιμετωπίζονται τα συντακτικά λάθη;

Τα σύγχρονα προγραμματιστικά περιβάλλοντα ανιχνεύουν εύκολα τα συντακτικά λάθη καθώς παρέχουν εργαλεία ελέγχου της σύνταξης των εντολών και προτείνουν, εμφανίζοντας ενημερωτικά πλαίσια, αναλυτικά τον τρόπο διόρθωσης τους. Ακόμα και αν δεν εντοπιστούν από το προγραμματιστικό περιβάλλον εντοπίζονται από τον **μεταγλωττιστή**, ο οποίος εμφανίζει προς τον προγραμματιστή κάποιο προειδοποιητικό μήνυμα. Αν ένα πρόγραμμα έχει συντακτικό λάθος *δεν εκτελείται* μέχρι να διορθωθεί από τον προγραμματιστή.

### 40. Ποιά λάθη λέγονται Αντικανονικού Τερματισμού;

Τα λάθη αυτά, που ονομάζονται και **λάθη εκτέλεσης**, εμφανίζονται κατά τη διάρκεια εκτέλεσης ενός προγράμματος. Είναι τα πιο επώδυνα, γιατί συνήθως εμφανίζονται σε πραγματικό χρόνο εκτέλεσης και προκαλούν αντικανονικό τερματισμό της εφαρμογής και το *κρέμασμα* του συστήματος.

#### Λάθη Αντικανονικού Τερματισμού:

- Διαίρεση με το μηδέν.
- Υπολογισμός ρίζας αρνητικού αριθμού, εφαπτομένης  $90^\circ$  και λογάριθμου αριθμού μικρότερου ή ίσου του μηδενός.
- Υπερχείλιση αριθμητικής μεταβλητής.
- Είσοδος (κατά το διάβαση) διαφορετικού τύπου δεδομένων.
- Χρήση μη αρχικοποιημένων μεταβλητών.
- Υπέρβαση των ορίων του πίνακα.

### 41. Πώς αντιμετωπίζονται τα λάθη Αντικανονικού Τερματισμού;

Όταν ένα λάθος προκληθεί κατά την εκτέλεση της εφαρμογής είναι δύσκολο να αντιμετωπιστεί. Οι περισσότερες σύγχρονες γλώσσες προγραμματισμού προσφέρουν τρόπους διαχείρισης τέτοιων λαθών με τη χρήση εντολών του προγράμματος που το παγιεύουν και στη συνέχεια εκτελούν τις κατάλληλες διαδικασίες χειρισμού του.

### 42. Πώς προλαμβάνονται και που οφείλονται τα λάθη αντικανονικού τερματισμού (ή λάθη κατά την εκτέλεση);

Η πρόληψη τέτοιων λαθών είναι δύσκολη, εφόσον συνήθως οφείλονται σε καταστάσεις που δεν είναι εύκολο να ελεγχθούν από τον προγραμματιστή, ενώ πολλές φορές εμφανίζονται μετά από ένα μεγάλο χρονικό διάστημα. Τέτοια λάθη προκύπτουν από

την κλήση μιας διαδικασίας με δεδομένα που δεν μπορεί να χειριστεί, όπως η αναζήτηση διαγραμμένων αρχείων, η προσπάθεια διαίρεσης ενός αριθμού με το μηδέν, η υπερχείλιση μιας αριθμητικής μεταβλητής, ή από δυσλειτουργία του υλικού μέρους του υπολογιστή, όπως η καταστροφή του σκληρού δίσκου, ή από τερματισμό σύνδεσης του δικτύου.

#### 43. Που οφείλονται και πώς αντιμετωπίζονται τα λογικά λάθη ενός προγράμματος;

Τα λογικά λάθη οφείλονται σε λανθασμένη σχεδίαση του προγράμματος. Όλες οι εντολές του, σε αντίθεση με τα συντακτικά λάθη ή τα λάθη εκτέλεσης, εκτελούνται κανονικά αλλά με εσφαλμένο αποτέλεσμα. Τα λάθη αυτά είναι τα πιο δύσκολο να εντοπιστούν καθώς δεν ανιχνεύονται ούτε από μεταφραστικά προγράμματα ούτε από κάποιο εργαλείο του υπολογιστή. Ο μόνος τρόπος να εντοπιστούν είναι με δοκιμαστικές εκτελέσεις του προγράμματος, διαδικασίες ελέγχου, που αναλύουν και συγκρίνουν τα αποτελέσματα του προγράμματος με τα αναμενόμενα.

##### Λογικά λάθη:

- Λανθασμένη εκτίμηση της ιεραρχίας των αριθμητικών τελεστών σε μια εκφραση.
- Λανθασμένη επιλογή τελεστών (π.χ. χρήση του και αντί του ή)
- Αρχικοποίηση μεταβλητής με λανθασμένη τιμή.
- Δομή επανάληψης που είναι ατέρμων βρόχος.

#### 44. Τι εννοούμε με τον όρο εκσφαλμάτωση προγράμματος;

Η διαδικασία ελέγχου, εντοπισμού και διόρθωσης των σφαλμάτων ενός προγράμματος καλείται **εκσφαλμάτωση** (debugging). Στόχος της διαδικασίας είναι ο εντοπισμός των σημείων του προγράμματος που προκαλούν προβλήματα στη λειτουργία του.

#### 45. Κατά πόσο είναι εύκολη η διαδικασία της εκσφαλμάτωσης (τι γνώσεις απαιτούνται για να γίνει η εκσφαλμάτωση);

Η εκσφαλμάτωση δεν είναι εύκολη διαδικασία, αλλά απαιτεί βαθιά γνώση της γλώσσας προγραμματισμού και ικανότητες από τη μεριά του προγραμματιστή. Η εκσφαλμάτωση είναι ένα πρόβλημα λογικής και όσο πιο καλά αντιλαμβάνεται ο προγραμματιστής τον τρόπο με τον οποίο λειτουργεί το πρόγραμμα, τόσο πιο εύκολα και σύντομα θα εντοπίσει τα λάθη.

#### 46. Ποιά λάθη μας απασχολούν κατά τη φάση της εκσφαλμάτωσης;

Τα λάθη που μας απασχολούν είναι τα λογικά και τα λάθη που παρουσιάζονται κατά την εκτέλεση ενός προγράμματος (αντικανονικού τερματισμού). Η εκσφαλμάτωση τέτοιων λαθών μπορεί να γίνει μέσα από εργαλεία εκσφαλμάτωσης ή από ειδικές εντολές ή συναρτήσεις τις οποίες προσφέρει το περιβάλλον προγραμματισμού. Τα συντακτικά λάθη δεν μας απασχολούν στη φάση της εκσφαλμάτωσης, εφόσον τις περισσότερες φορές το προγραμματιστικό περιβάλλον τα ανιχνεύει αυτόματα και προτείνει τη διόρθωσή τους.

#### 47. Πώς γίνεται ο έλεγχος ορθότητας και ο εντοπισμός λογικών λαθών ενός προγράμματος;

Για τον έλεγχο της ορθότητας ενός προγράμματος σχηματίζουμε *σενάρια ελέγχου*, δηλαδή πραγματοποιούμε δοκιμαστικές εκτελέσεις του προγράμματος χρησιμοποιώντας ως τιμές εισόδου, δεδομένα που αντιπροσωπεύουν όλες τις περιπτώσεις που ελέγχονται.

<b>Πώς αποτυπώνεται η εκτέλεση σεναρίων ελέγχου;</b> Κάθε σενάριο ελέγχου αντιστοιχεί σε μια δοκιμή με τις εισόδους και τα αποτελέσματα να αποτυπώνονται στον παρακάτω πίνακα.					
A/A	Τιμές εισόδου	Αναμενόμενο αποτέλεσμα	Αποτέλεσμα συνθήκης	Έξοδος Προγράμματος	Ορθότητα Αποτελέσματος

**48. Με τι σχετίζονται τα λάθη που εμφανίζονται στις δομές επιλογής;**

Σε μια δομή επιλογής, τα πιο συνηθισμένα λάθη σχετίζονται με:

- Τις εντολές πριν από την δομή, που επηρεάζουν τη συνθήκη της.
- Τη συνθήκη ή τις συνθήκες τις δομής
- Τις ομάδες εντολών που εκτελούνται όταν μια συνθήκη είναι αληθής ή ψευδής.

**49. Με τι σχετίζονται τα λάθη που εμφανίζονται στις δομές επανάληψης;**

Σε μια δομή επανάληψης, τα πιο συνηθισμένα λάθη σχετίζονται με:

- Τη συνθήκη επανάληψης ή τερματισμού.
- Την αρχικοποίηση της συνθήκης.
- Την ενημέρωση της συνθήκης εντός του βρόχου
- Τις εντολές που περιλαμβάνονται εντός του βρόχου.

**50. Με τι σχετίζονται τα λάθη που εμφανίζονται στους πίνακες;**

Κατά την εκσφαλμάτωση προγραμμάτων που χρησιμοποιούν πίνακες, χρειάζεται να δίνεται ιδιαίτερη προσοχή:

- Στο μέγεθος των πινάκων κατά τη δήλωσή τους.
- Στους δείκτες των πινάκων κατά την προσπέλασή τους.
- Στη μη υπέρβαση των ορίων των πινάκων.

**51. Με τι σχετίζονται τα λάθη που εμφανίζονται στα υποπρογράμματα;**

Κατά την εκσφαλμάτωση προγραμμάτων που χρησιμοποιούν υποπρογράμματα, χρειάζεται να δίνεται προσοχή στον εντοπισμό των λογικών λαθών που σχετίζονται με:

- Την κλήση υποπρογράμματος και το πέρασμα των παραμέτρων.
- Τα λοιπά λογικά λάθη που εμφανίζονται στο πρόγραμμα.