

Λειτουργικά Συστήματα (HY321)

Διάλεξη 11:
Δίσκοι





Πώς είναι;

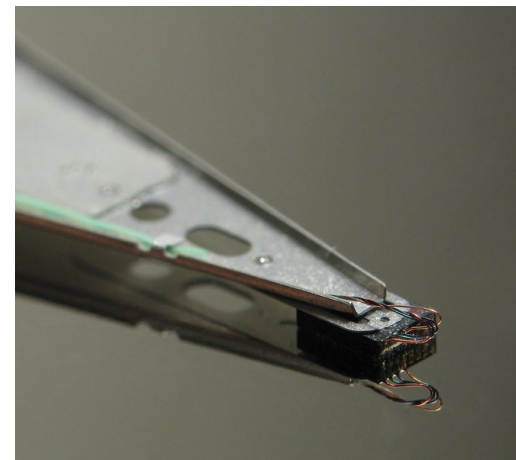
- Η πρώτη εντύπωση...
- Η γυμνή αλήθεια...
 - Και ναι, όντως, υπάρχει μέσα δίσκος
 - Και ναι, είναι και σκληρός





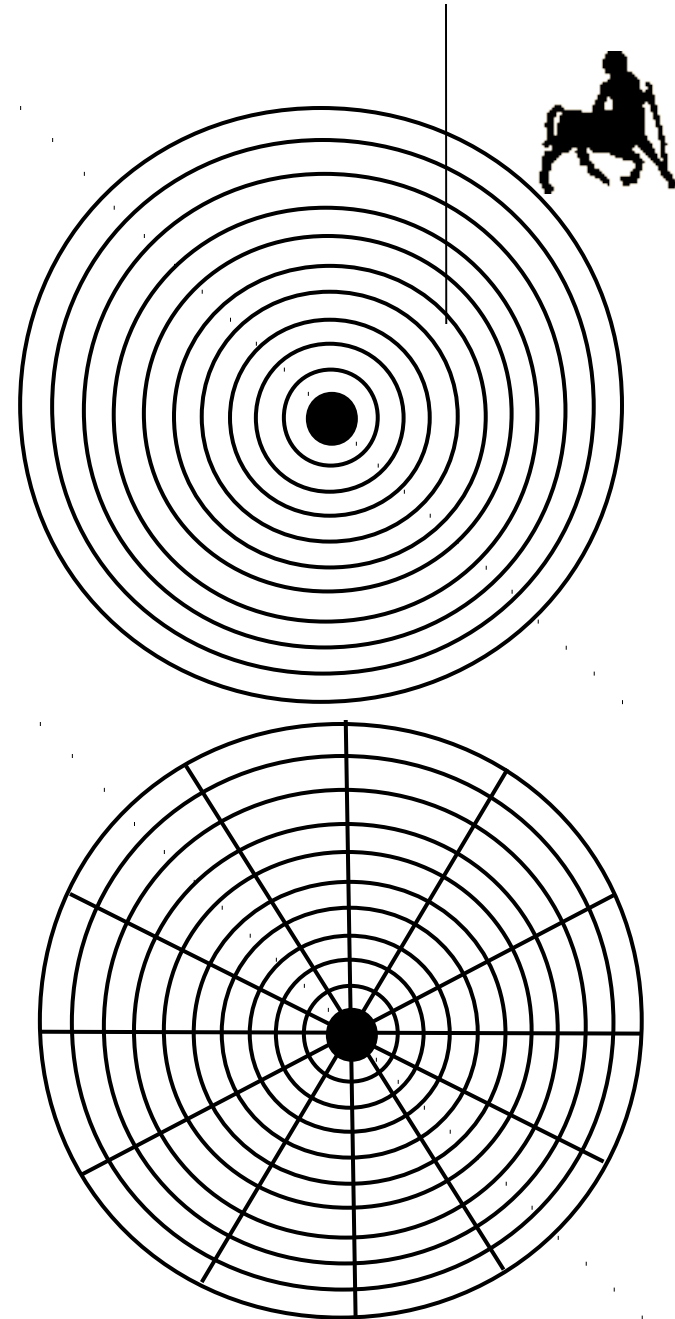
Ας Γνωριστούμε Καλύτερα...

- Κατ' ακρίβεια έχει πολλούς «σκληρούς» δίσκους μέσα...
 - Τον έναν πάνω από τον άλλον
 - Γυρίζουν γρήγορα
 - 5400, 7200, 10000, 15000 rpm
 - 1-15 δίσκοι
- Πάνω (και κάτω) από τον καθένα υπάρχουν «κεφαλές»
 - Διαβάζουν και γράφουν
 - «Επιπλέουν» πάνω σε ένα λεπτό στρώμα αέρα
 - 2-30 κεφαλές (2 ανά δίσκο)



Οργάνωση

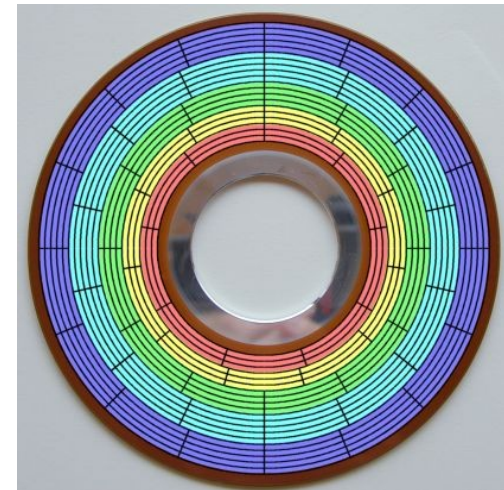
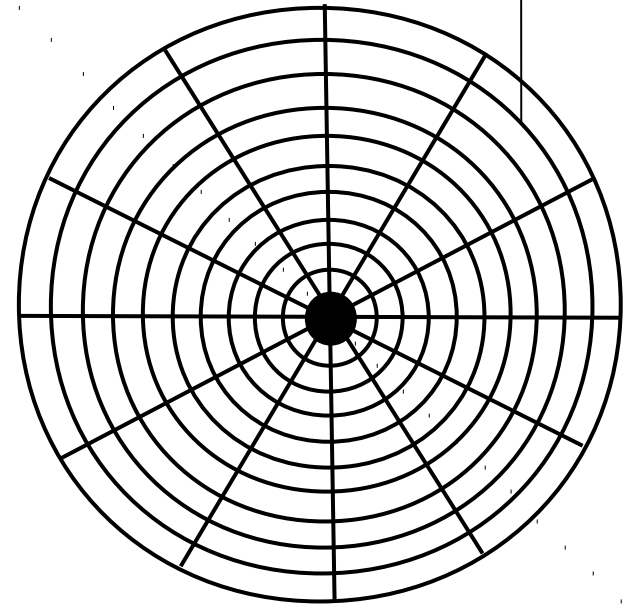
- Πώς οργανώνεται κάθε επιφάνεια;
 - Ομόκεντροι κύκλοι (διαδρομές – tracks)
 - 700-20480 διαδρομές/επιφάνεια
 - Τομείς (sectors) σε κάθε διαδρομή
 - Ορίζονται μεταξύ δύο ομόκεντρων κύκλων και μεταξύ των πλευρών μιας συγκεκριμένης γωνίας
 - 16-1800 τομείς/διαδρομή
 - 64-8K bytes/τομέα
 - Συνήθως 512 bytes/τομέα
 - Το μικρότερο κομμάτι που μπορεί να γραφτεί/διαβαστεί από το δίσκο





Οργάνωση

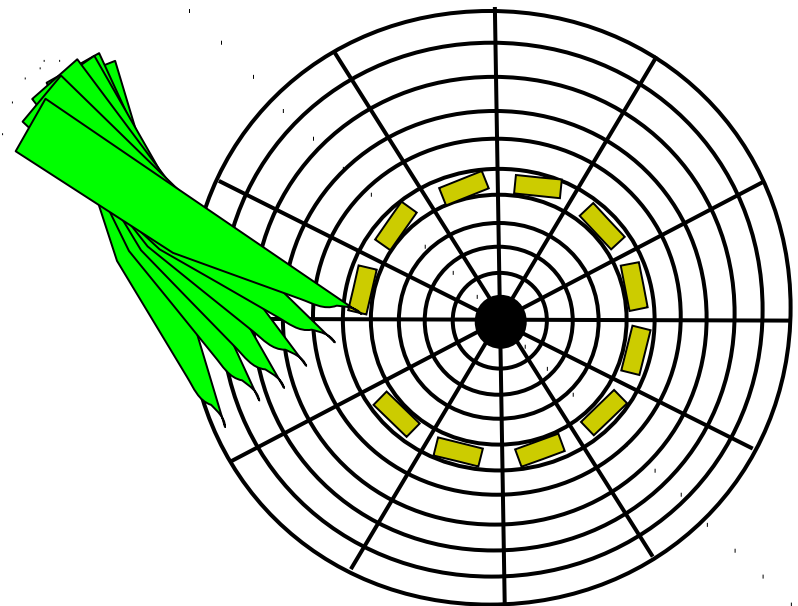
- Εξωτερικοί τομείς:
 - Λίγο μεγαλύτεροι;
 - Περισσότερη επιφάνεια / bit
 - Μεγαλύτερη αξιοπιστία (κάποια ΛΣ το αξιοποιούν)
 - Αλλά και χαμένος χώρος.
 - Εναλλακτική;
 - Πολλαπλές ζώνες
 - Διαφορετική «γωνία» ανά τομέα για κάθε ζώνη
 - Όπως κινείται η κεφαλή προσαρμόζεται και η ταχύτητα περιστροφής
 - Περίπου ίδια μεγέθη τομέων



Ανάγνωση / Εγγραφή από Τομέα



- Πρέπει να συμβούν τα εξής:
 - **Μετακίνηση του βραχίονα** πάνω στη σωστή διαδρομή
 - **Χρόνος αναζήτησης (seek time)**
 - Χειρότερη περίπτωση ~30 – 50 msec (διέτρεξε όλο το δίσκο)
 - Καλύτερη περίπτωση 0 msec (πάνω από τη σωστή διαδρομή)
 - Μέση περίπτωση ~10-17 msec (όσο το 1/3 της χειρότερης)
 - **Αναμονή** έως ότου ο τομέας φτάσει (λόγω της περιστροφής του δίσκου) κάτω από την κεφαλή
 - **Καθυστέρηση περιστροφής (rotational latency)**
 - Χειρότερη περίπτωση ~6 msec (ολόκληρη περιστροφή, 10000 rpm)
 - Μέση περίπτωση ~3msec (μισή περιστροφή, 10000 rpm)
 - Καλύτερη περίπτωση 0msec (πάνω από το σωστό τομέα)



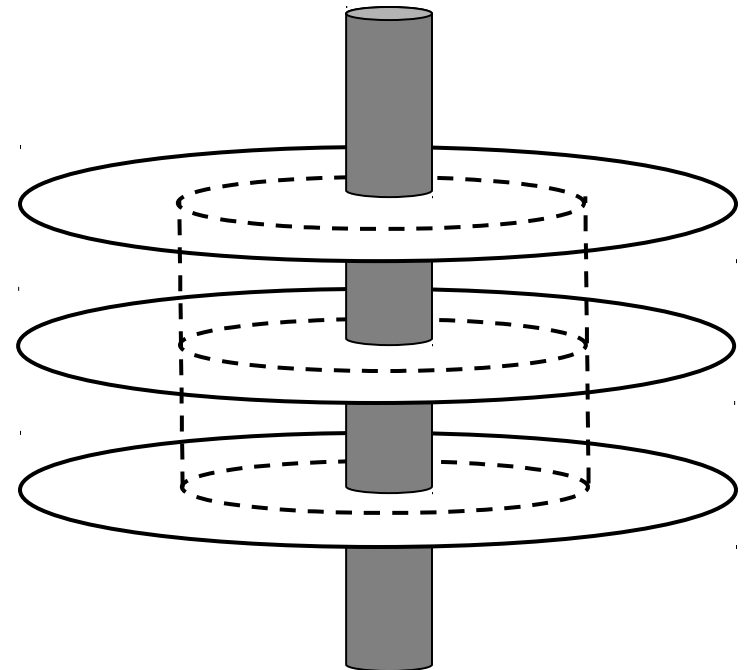
It's Show-Time!!!





Ο Κύλινδρος

- Οι ίδιες διαδρομές στις διαφορετικές επιφάνειες σχηματίζουν έναν «κύλινδρο»





Χρόνος Προσπέλασης

- Χρόνος αναζήτησης (μέσος):
 - 10-17 msec
- Καθυστέρηση περιστροφής (μέσος)
 - 3 msec
- Και κάτι ψιλά ακόμα:
 - Για να σταθεροποιηθεί η κεφαλή πάνω από τη διαδρομή
 - Για να επεξεργαστεί ο δίσκος την εντολή
- Χρόνος προσπέλασης (μέσος):
 - ~20 msec (άρα κατά μέσο όρο 50 προσπελάσεις / sec)
- Ρυθμός μεταφοράς ~ 50 μεταφορές/sec * 512 bytes/μεταφορά (1 τομέας) = ~25600 bytes/sec
 - Αργόoooooooo (πιο αργό και από την Dial-up)
 - Για μισό λεπτό: Οι κατασκευαστές μιλάνε για μεταφορές Mbytes/sec
 - Τους πιάσαμε;
 - Όχι ακριβώς

Καθυστέρηση και Ρυθμός Μεταφοράς (Latency / Bandwidth)

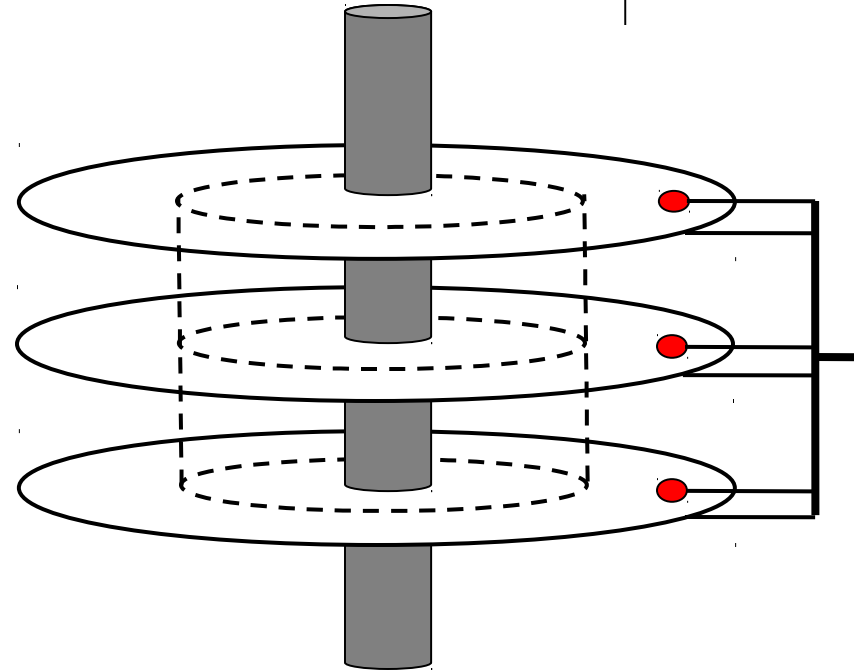


- Καθυστέρηση
 - Χρόνος για να ξεκινήσει μια μεταφορά
- Χρόνος για να μεταφέρουμε μια ολόκληρη διαδρομή;
 - Καθυστέρηση + Χρόνος για 1 περιστροφή (6msec)
 - Γιατί;
 - Χρόνος για να μεταφέρουμε 1 ολόκληρο κύλινδρο;
- Ρυθμός μεταφοράς
 - Ο ρυθμός με τον οποίο μεταφέρονται τα δεδομένα αφότου ξεκινήσει η μεταφορά
 - Για δίσκο 10000rpm με 1024 τομείς/διαδρομή, 512 bytes/τομέα, ποιο είναι το μέγιστο bandwidth;
 - Για μια περιστροφή: 6msec
 - Σε μία περιστροφή $1024 * 512 \text{ bytes} = 512 \text{ KBytes}$
 - Άρα περίπου 83,3 Mbytes/sec

«Φθηνή» Προσπέλαση στον Ίδιο Κύλινδρο!



- Όλες οι κεφαλές στον ίδιο βραχίονα
 - Όλες οι κεφαλές στον ίδιο κύλινδρο
- Φθηνή «εναλλαγή» ενεργής κεφαλής
 - Σβήσε τη 0, άναψε την 1
 - Σβήσε την 1, άναψε τη 2
 - ...
- Βέλτιστος ρυθμός μεταφοράς:
 - Μετέφερε όλους τους τομείς μιας διαδρομής
 - Μετέφερε όλες τις διαδρομές ενός κυλίνδρου
 - Μετακίνησε την κεφαλή σε άλλο κύλινδρο





Ενδιαφέρουσες Τάσεις

- Το εύρος ζώνης του δίσκου και το κόστος/bit μειώνονται εκθετικά
 - Παρόμοια με την ταχύτητα CPU, το μέγεθος μνήμης...
- Ο χρόνος αναζήτησης και η καθυστέρηση περιστροφής μειώνονται πολύ αργά!
 - Γιατί;
 - Απαιτούν φυσική μετακίνηση (δίσκος, βραχίονας)
- Συνέπειες:
 - Προσπέλαση δίσκου: Σημείο μεγάλης συμφόρησης (και χειροτερεύει)
 - Αφού μεγαλώνει το εύρος ζώνης μπορούμε να προ-μεταφέρουμε μεγάλα κομμάτια δεδομένων στο κόστος περίπου ενός μικρού κομματιού
 - Άρα; Αντάλλαξε εύρος ζώνης με ταχύτητα όταν μπορείς να μεταφέρεις σχετιζόμενα δεδομένα
 - Ομαδοποίησε τα δεδομένα στο δίσκο
 - Το μέγεθος της μνήμης αυξάνει γρηγορότερα από το λειτουργικό σύνολο των τυπικών εφαρμογών
 - Χώρος για caching
 - Η κίνηση προς το δίσκο αλλάζει: Κυρίως εγγραφές και νέα δεδομένα



Τι να Κάνω Λοιπόν;

- Μετέφερε πολλά μαζί
 - Μεταφορές πολλαπλών τομέων κάθε φορά
- Μη χοροπηδάς από 'δω και από 'κει
 - Χρονοδρομολόγηση της κίνησης του βραχίονα

Αλγόριθμοι Δρομολόγησης (Βραχίονα) Δίσκου



- Φέρσου καλά στο δίσκο
 - Για να σε ανταμοίψει...
- Πώς;
 - Δώσε του **αιτήσεις** για αναγνώσεις/εγγραφές **σε κοντινά σημεία**
 - Μειώνει το χρόνο αναζήτησης
 - Και ίσως και την καθυστέρηση περιστροφής



Αναφορές Στο Δίσκο

- Τι ξέρει το ΛΣ;
 - Είδος (IDE/SCSI/SATA), αριθμός συσκευής, αριθμός κυλίνδρων/κεφαλών/τομέων (**CHS**)
 - OK, αυτό ίσχυε στους παλιούς δίσκους
 - Στους καινούριους «διευθυνσιοδοτούμε» με έναν «αφηρημένο» αριθμό τομέα
 - **LBA**: Logical Block Addressing
 - Ποιος χρησιμοποιεί τους αριθμούς τομέα;
 - Τα συστήματα αρχείων
 - Αναθέτουν τομείς σε αρχεία
 - Και μια ασυνέπεια ορολογίας:
 - Diskάδες: Block \Leftrightarrow τομέας (sector)
 - Filesystemάδες: Block \Leftrightarrow ομάδα πολλαπλών συνεχών τομέων

«Διευθύνσεις» στο Δίσκο και Δρομολόγηση

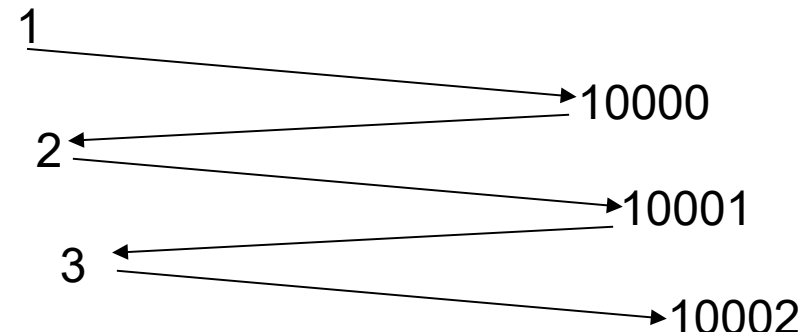


- Στόχος δρομολογητή δίσκου
 - Ουρά αιτήσεων
 - Όταν ολοκληρώνεται μία αίτηση διάλεξε την πιο «κοντινή» από την ουρά
- Στόχος «γραμμικών διευθύνσεων» τομέων
 - **Απόκρυψη** των «βρώμικων» λεπτομερειών σχετικά με τη θέση κάθε τομέα στο δίσκο
- Η θέση του ΛΣ
 - Παλαιότερα το ΛΣ ήθελε να ξέρει τη διάταξη του δίσκου
 - Τώρα προσπαθεί να μην ασχολείται
 - Υποθέτει ότι κοντινές γραμμικές διευθύνσεις βρίσκονται κοντά στο δίσκο
 - Κάποια ερευνητικά ΛΣ αρχίζουν πάλι να ασχολούνται
 - Εμείς δε θέλουμε να ξέρουμε...

FCFS (Εξυπηρέτηση με τη Σειρά Άφιξης)



- Προφανές...
- Πλεονεκτήματα;
 - Απλό
 - Αρκεί μια ουρά με τις αιτήσεις στον οδηγό του δίσκου
 - Δίκαιο
- Μειονεκτήματα;
 - Μεγάλα άλματα πάνω στο δίσκο
 - Και άρα και αντίστοιχα μεγάλος χρόνος αναζήτησης (και απόκρισης)
 - Άνευ λόγου και αιτίας
 - Παράδειγμα: 1, 10000, 2, 10001, 3, 10002



Πρώτα η Κοντινότερη Αναζήτηση (Shortest Seek Time First – SSTF)



- Εξυπηρέτησε τις αιτήσεις που είναι πιο **κοντά** στην τρέχουσα θέση του βραχίονα
 - Υπολόγισε αφαιρώντας τους αριθμούς των τομέων
 - Το αντίστοιχο του **SJF** (από τη χρονοδρομολόγηση διεργασιών)
 - Επειδή εδώ ξέρουμε ποιο είναι το shortest job
- Πλεονεκτήματα:
 - Μικρός μέσος χρόνος αναζήτησης
 - Καλός ρυθμός εξυπηρέτησης
- Μειονεκτήματα:
 - Υπερβολικά ψηλή απόκλιση στο χρόνο απόκρισης
 - Κάποιες αιτήσεις μπορεί να «λιμοκτονήσουν»
 - Πότε;

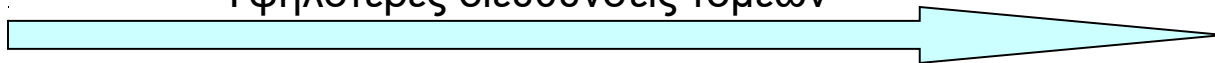


SSTF

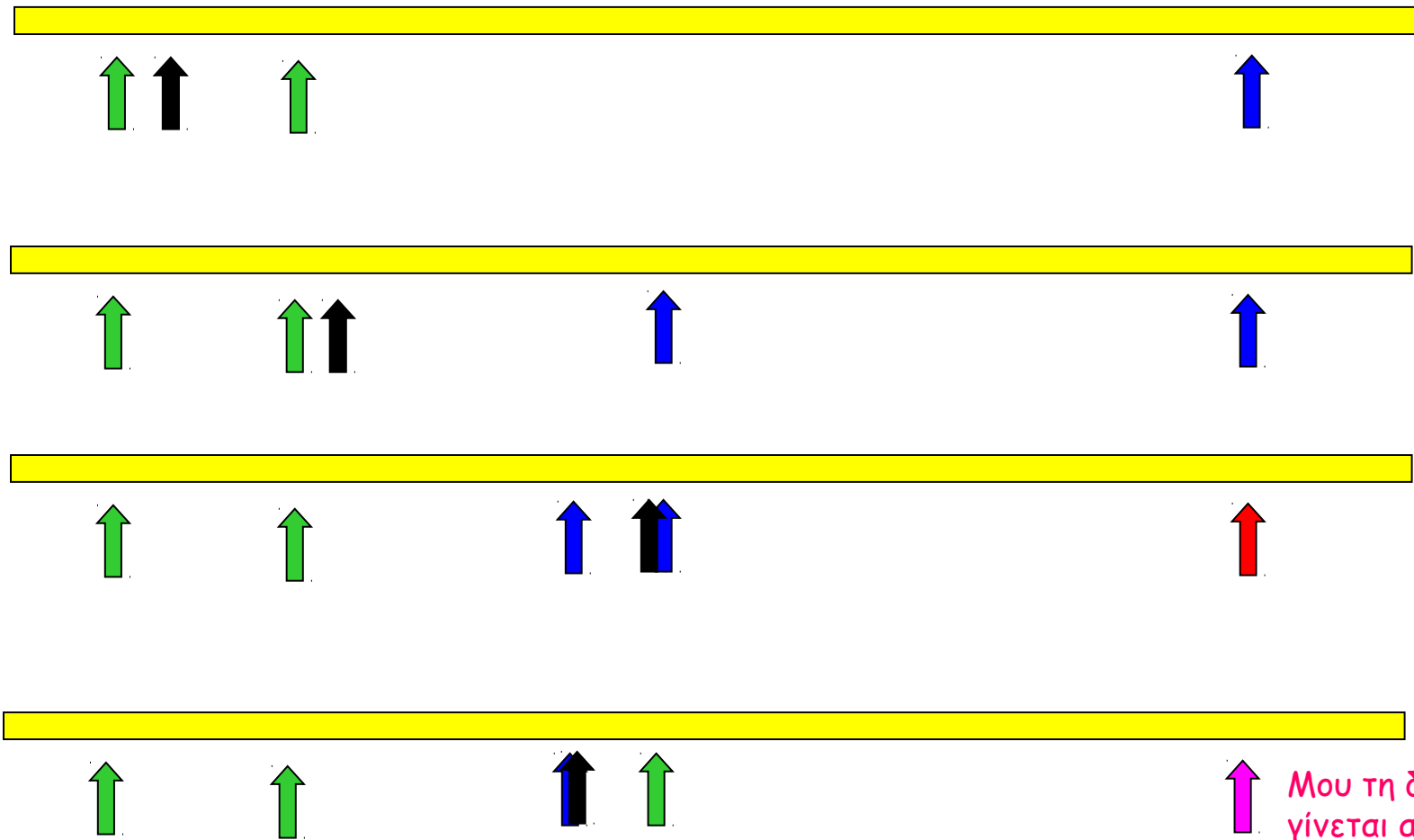
- Μαύρο: κεφαλή
- Μπλε: αιτήσεις
- Πράσινο: εξυπηρετηθείσες αιτήσεις
- Κόκκινο: «εκνευρισμένες» αιτήσεις
- Μωβ: αιτήσεις εκτός εαυτού...

SSTF

Υψηλότερες διευθύνσεις τομέων



SSTF



Μου τη δίνει όταν γίνεται αυτό!!!



Τι «Στράβωσε»;

- FIFO
 - Δίκαιος αλλά «βαρύς»...
 - Παραείναι δίκαιος
- SSTF
 - Αποδοτικός
 - Παραείναι αποδοτικός
 - Κοιτάζει μόνο κοντά στην κεφαλή
 - Ξεχνάει ότι κάποια στιγμή πρέπει να **σαρώσουμε** όλο το δίσκο
 - Άδικος και απρόβλεπτος! ☹
- Τι είπαμε ξεχνάει ο SSTF; Ας κάνουμε ακριβώς αυτό
 - Ας **σαρώσει** ο βραχίονας το δίσκο προς μία κατεύθυνση
 - Ας εξυπηρετήσουμε ότι αιτήσεις βρίσκουμε μπροστά μας
 - Μόλις ο βραχίονας «βρει» (κέντρο η άκρη) **αλλάζει κατεύθυνση**



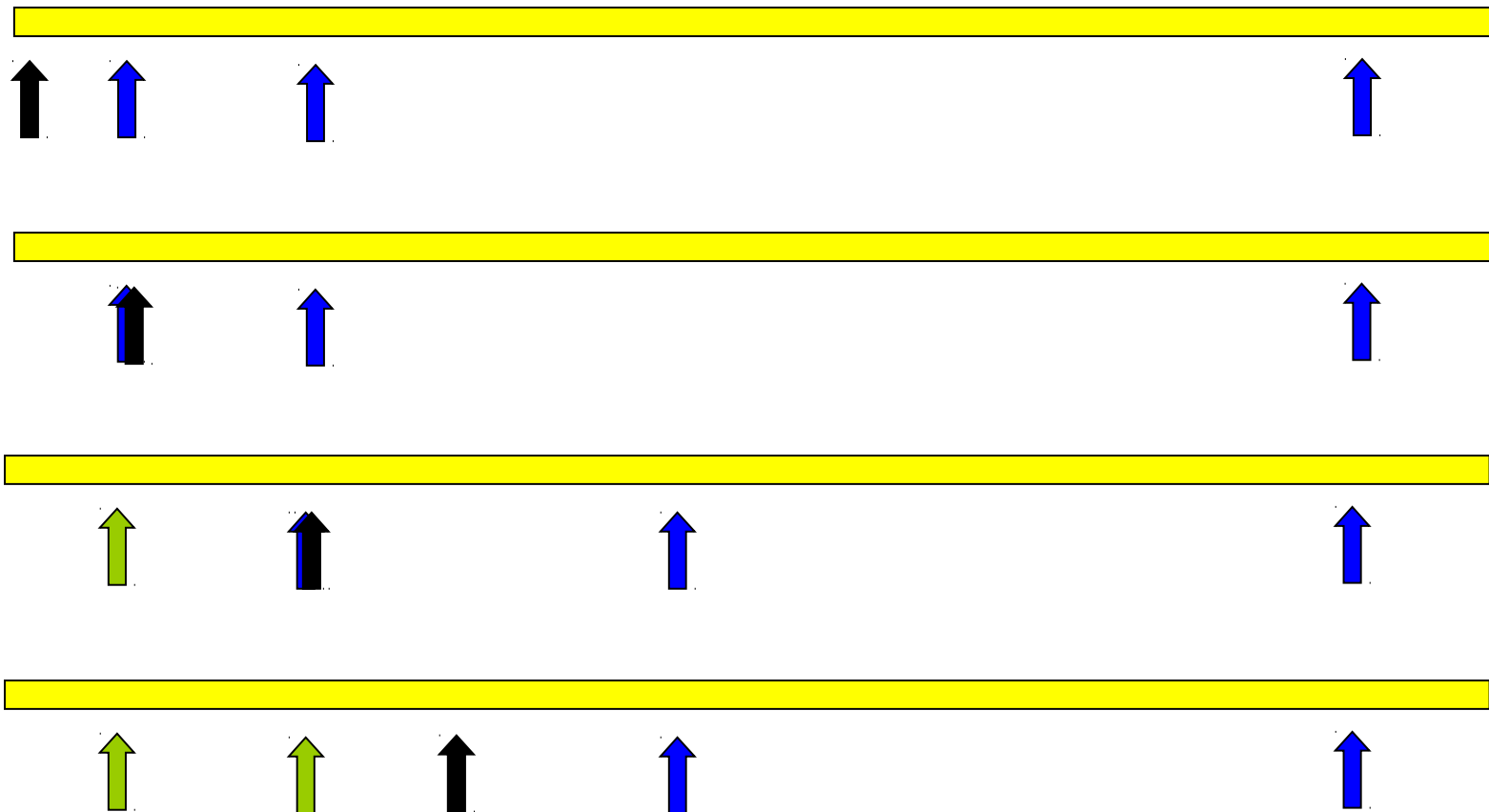
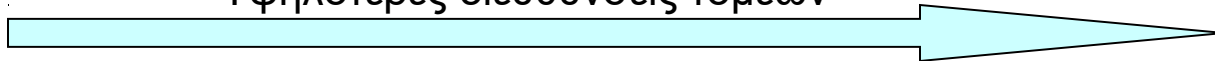
Αλγόριθμος SCAN

- Ουρά αιτήσεων
 - Λίστα (2πλά συνδεδεμένη)
 - Ταξινομημένη (με βάση τους αριθμούς των τομέων)
- Αλγόριθμος
 - κατεύθυνση = +1 ή -1
 - $\text{τρέχων_κύλινδρος} = \text{τρέχων_κύλινδρος} + \text{κατεύθυνση}$
 - Κοίτα την ουρά για αιτήσεις στον τρέχοντα κύλινδρο
 - Αν βρεις κάποιες, εξυπηρέτησέ τις
 - Αν $\text{τρέχων_κύλινδρος} == 0$ ή $\text{τρέχων_κύλινδρος} == \text{MAX}$
 - $\text{κατεύθυνση} = -\text{κατεύθυνση}$

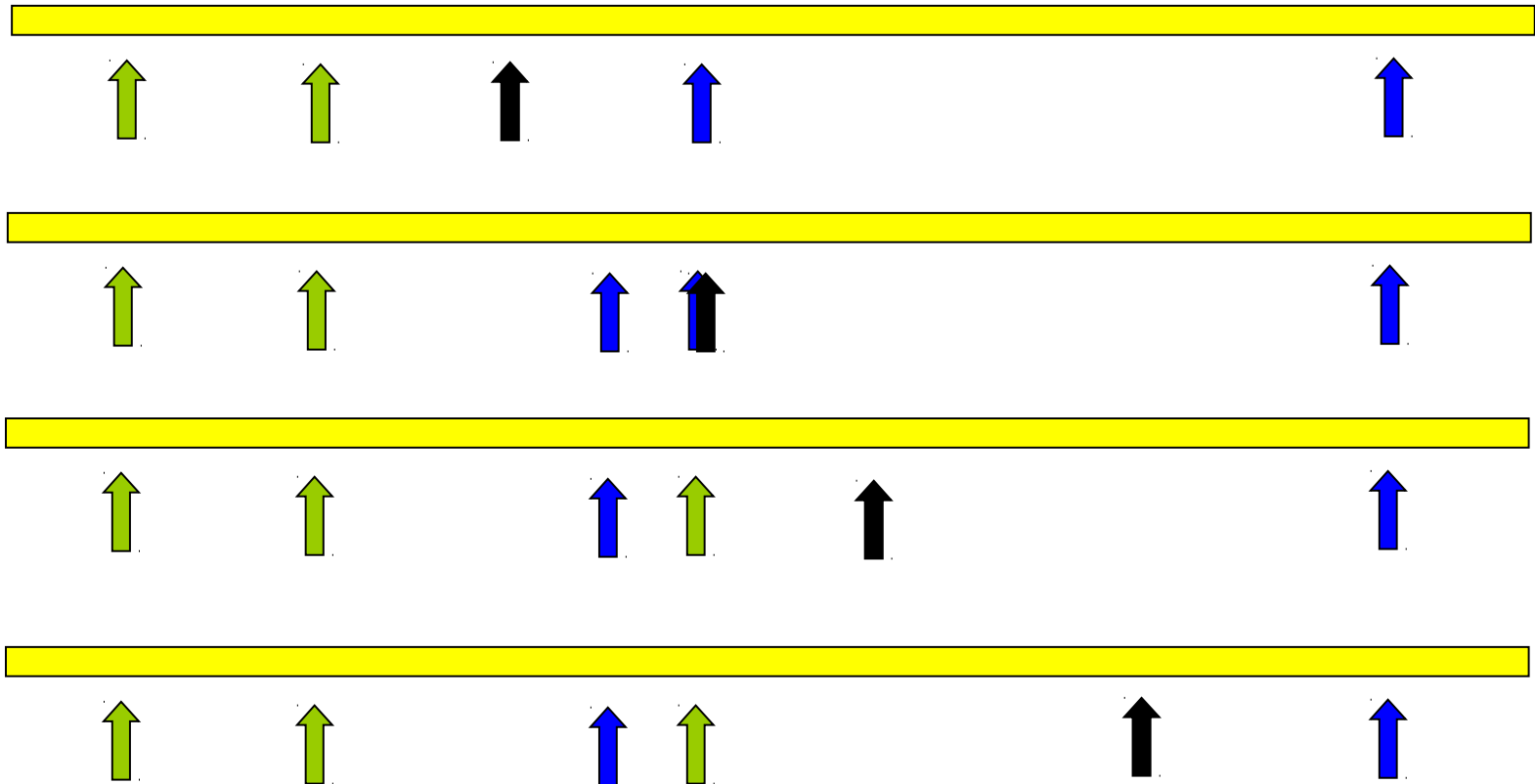
SCAN



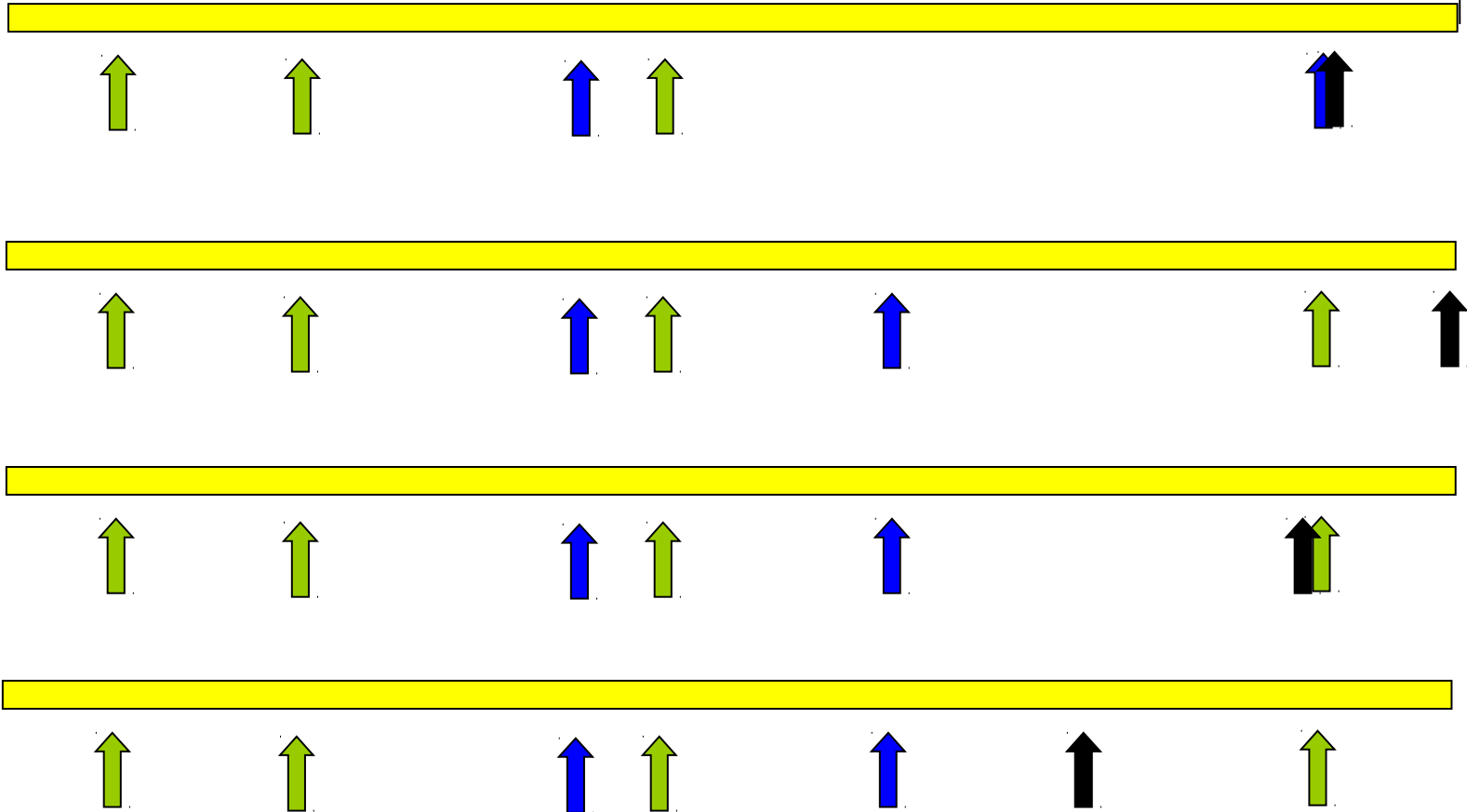
Υψηλότερες διευθύνσεις τομέων



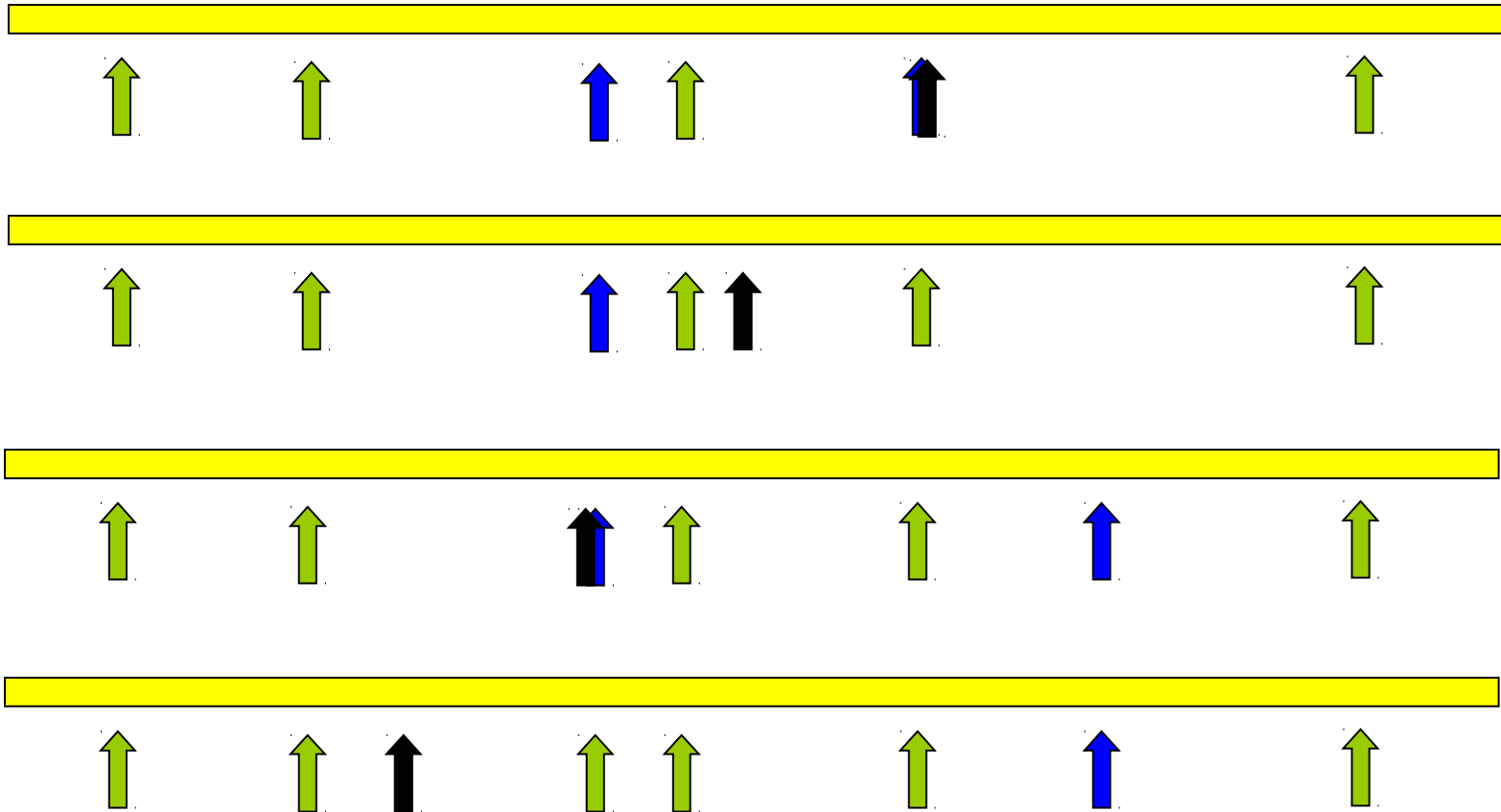
SCAN



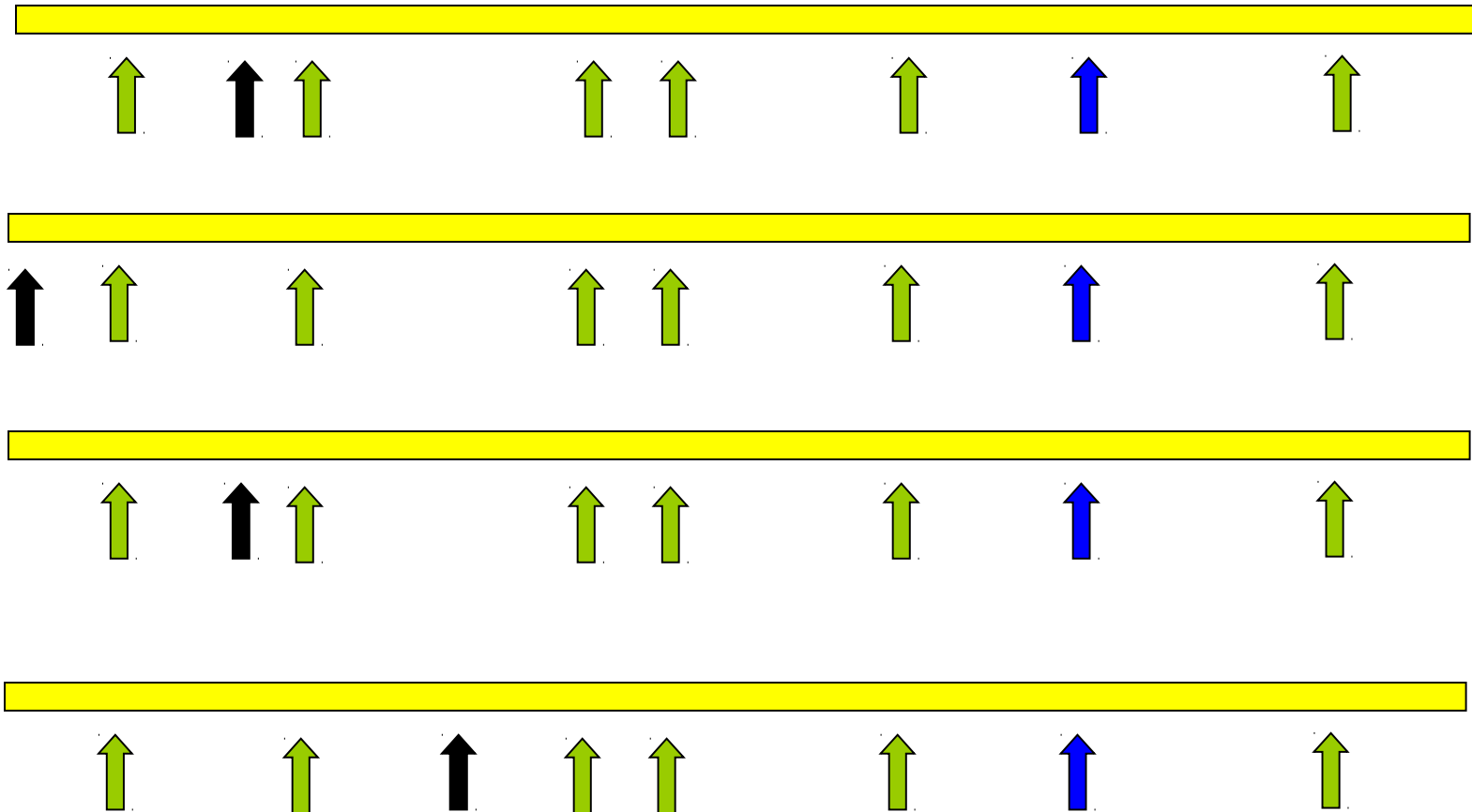
SCAN



SCAN



SCAN





Αλγόριθμος SCAN

- Τι καταφέραμε;
 - Μέσος χρόνος απόκρισης:
 - Καλύτερος από του FIFO
 - Χειρότερος από του SSTF
 - Απόκλιση χρόνου απόκρισης
 - Καλύτερη από του SSTF
- Παρεμπιπτόντως...
 - Χρειάζεται να πηγαίνουμε μέχρι τα άκρα του δίσκου;



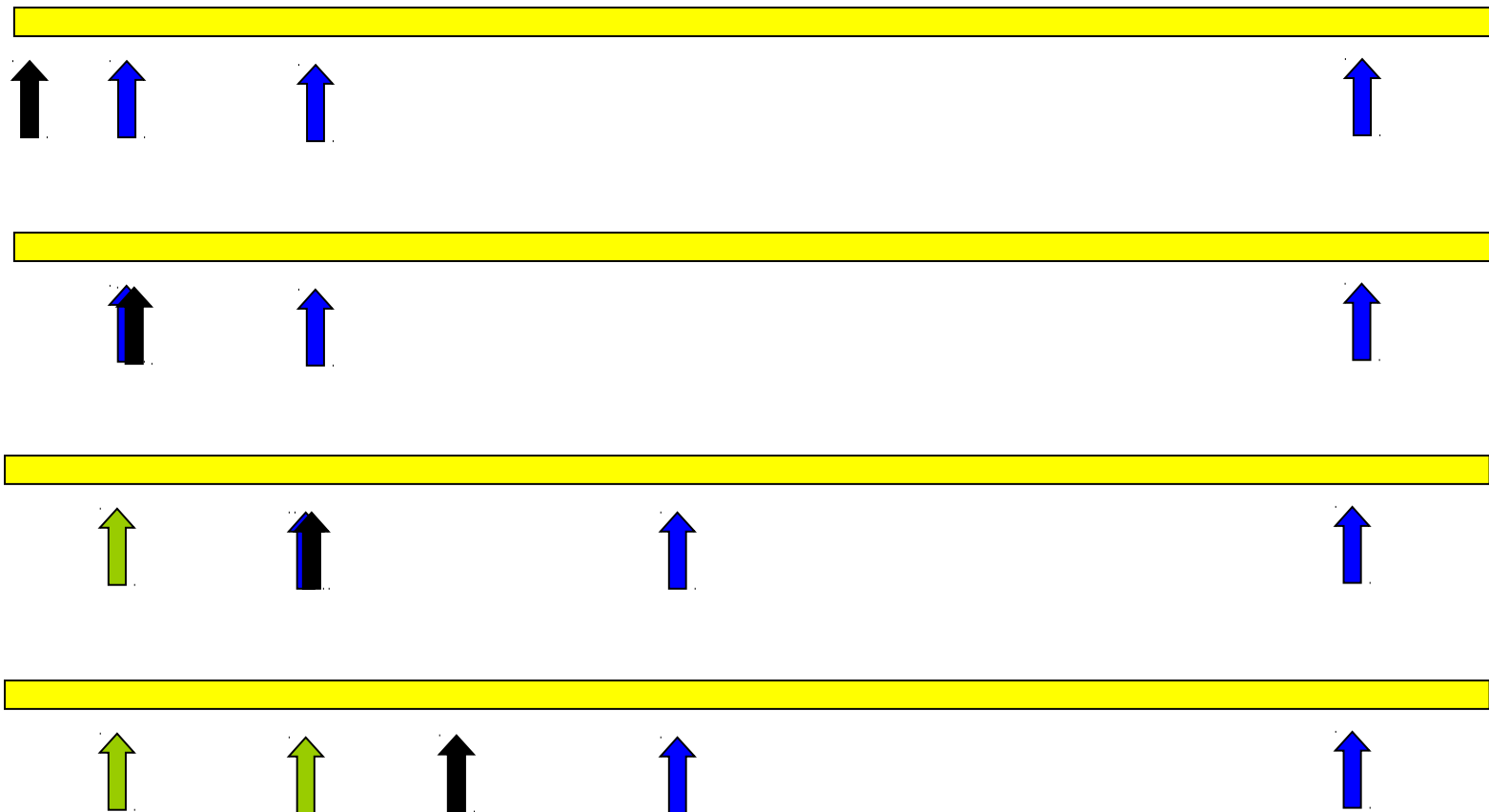
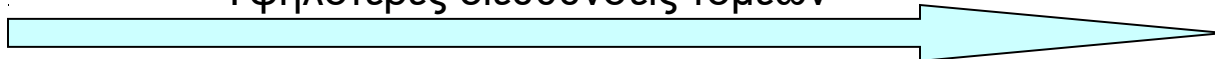
Αλγόριθμος LOOK

- Όπως ο SCAN
 - Απλά μην περιμένεις να ακούσεις το γκντουπ (κέντρο ή περιφέρεια) για να αλλάξεις κατεύθυνση
 - **Ανέστρεψε όταν δεν υπάρχουν άλλες αιτήσεις** προς την κατεύθυνση κίνησης του δίσκου
 - Βελτιώνει και το μέσο χρόνο απόκρισης και την απόκλισή του

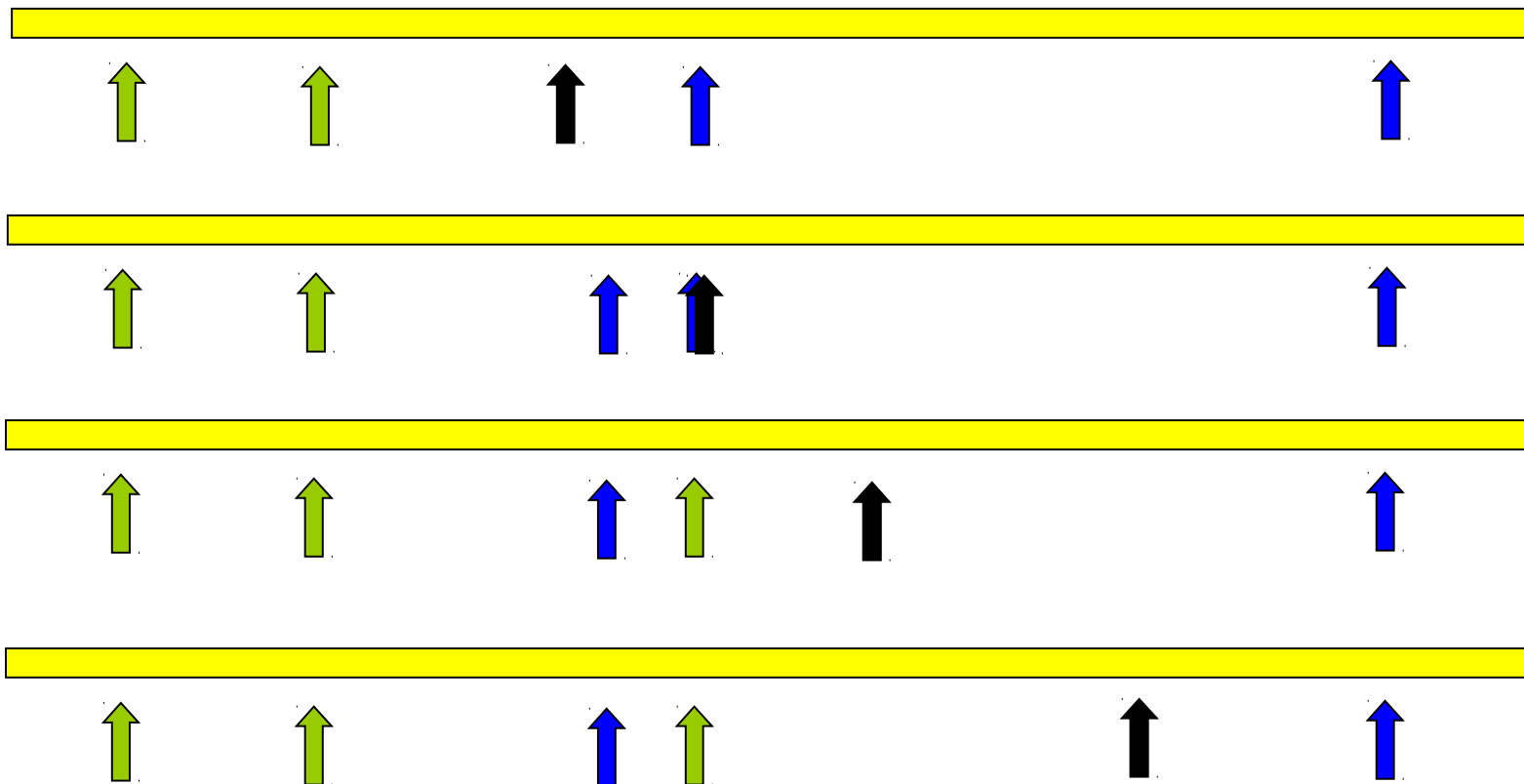
LOOK



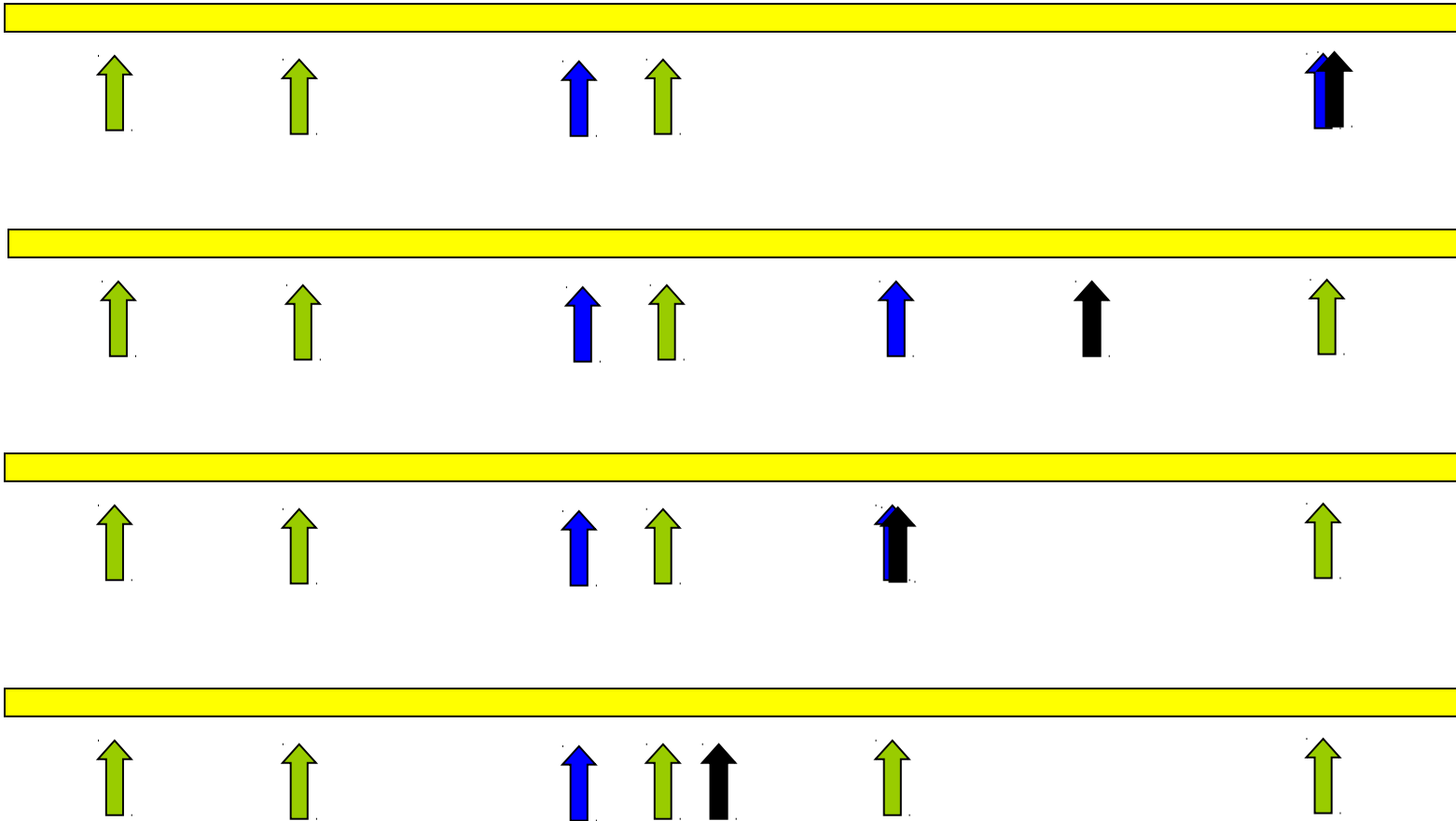
Υψηλότερες διευθύνσεις τομέων



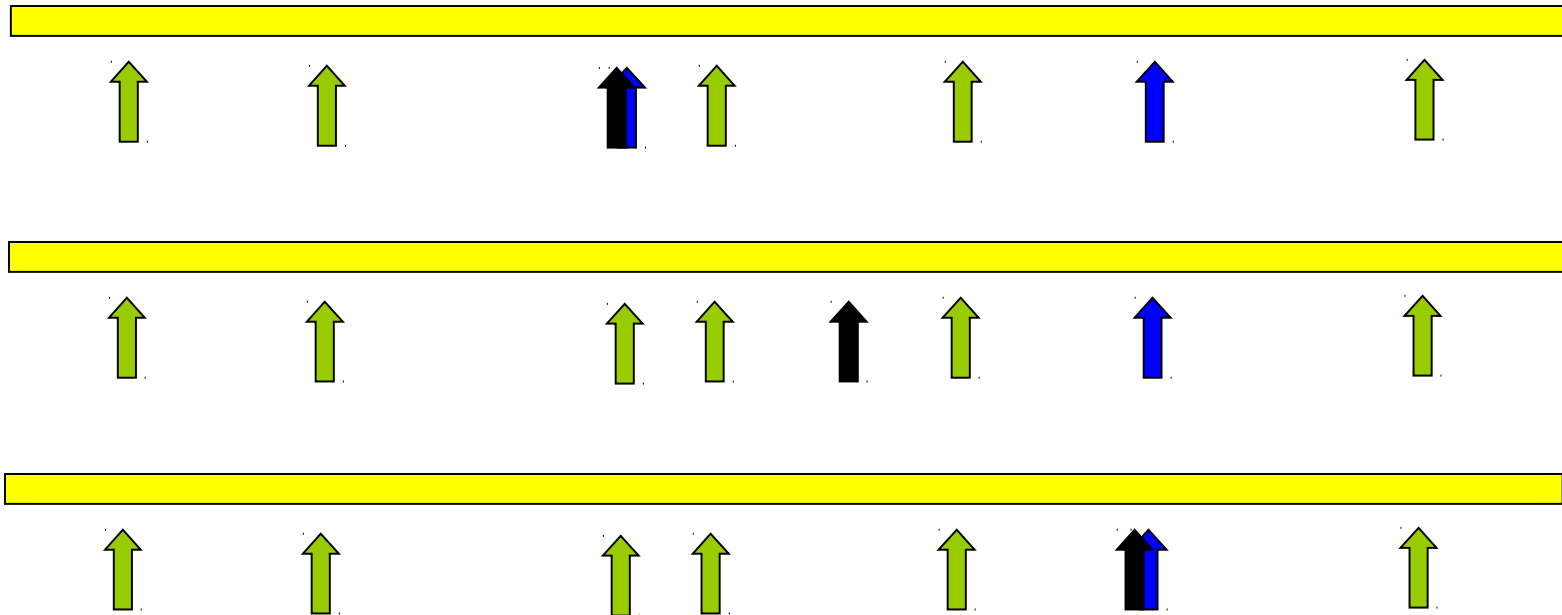
LOOK



LOOK



LOOK





C-SCAN (Κυκλικό SCAN)

- Οι SCAN και LOOK είναι λίγο άδικοι
 - Γιατί;
 - Εξυπηρετούν τους μεσαίους τομείς του δίσκου με διπλάσια συχνότητα από τους ακραίους
- C-SCAN
 - Στείλε τις αιτήσεις με αυξανόμενη σειρά κυλίνδρων
 - Όταν φτάσεις στο τέλος του δίσκου, επέστρεψε το βραχίονα στην αρχή του δίσκου
 - Μεγάλος χρόνος αναζήτησης
 - Όμως τον «φορτώνονται» πολλές αιτήσεις (όσες εξυπηρετούνται σε κάθε πέρασμα)
 - Επανέλαβε
 - Πλεονεκτήματα:
 - Δίκαιος
 - Βελτιώνεται ακόμα περισσότερο η απόκλιση του μέσου χρόνου απόκρισης



C-LOOK

- Ξεχάσαμε τίποτα;
 - Τα γνωστά...
 - Μην πηγαίνεις μέχρι το άκρο του δίσκου
 - Προχώρα μόνο όσο έχεις μπροστά σου αιτήσεις
 - Μην επιστρέφεις στην αρχή
 - Πήγαινε απλά στην πίσω-πίσω αίτηση
- Μεγάλο σουξέ...



Προβλημάκια...

- Έχουμε φτιάξει και ΤΟΝ δρομολογητή δίσκου;
 - Όχι ακριβώς...
- Ο χρόνος αναζήτησης είναι σημαντικός
 - Η C-LOOK τα πάει καλά...
- Η **καθυστερήση περιστροφής** όμως είναι συγκρίσιμη
 - Μια ολόκληρη περιστροφή δίσκου πολύ πιο αργή από μικρές αναζητήσεις
 - Αυτό που έχει σημασία είναι ο **χρόνος τοποθέτησης** (πάνω από το σωστό τομέα) και όχι ο χρόνος αναζήτησης

Πρώτα ο Μικρότερος Χρόνος Τοποθέτησης (Shortest Positioning Time First – SPTF)



- SPTF παρόμοιος με τον SSTF
 - Απλά αντί για τον κοντινότερο κύλινδρο, εξυπηρετούμε τον «**ΚΟΝΤΙΝΟΤΕΡΟ**» **τομέα**
- Άντε βρες...
 - Ποιος είναι ο κοντινότερος τομέας;
 - Δύσκολο μόνο από το γραμμικό αριθμό τομέα
 - Χρειάζεται η γεωμετρία του δίσκου και η τρέχουσα θέση (γωνία) των επιφανειών από κάποιο σημείο αναφοράς
- Καλύτερη επίδοση από την SPTF
 - Πάλι όμως μπορεί αιτήσεις να λιμοκτονήσουν



Ζυγισμένος SPTF (WSPTF)

- Βάλε δικαιοσύνη στον SPTF
- Χρόνος αναμονής στις αιτήσεις
 - Ψάξε την κοντινότερη αίτηση, ...
 - ... βάλε όμως και έναν παράγοντα γήρανσης που έχει να κάνει με το χρόνο αναμονής
 - Άρα... καμιά φορά εξυπηρέτησε μια παλιά αίτηση αντί για την κοντινότερη αίτηση.
- Διάφορες πολιτικές για τον παράγοντα γήρανσης
- Δουλεύει μια χαρά!
- Δύσκολο για το ΛΣ
 - Δεν ξέρει την ακριβή κατάσταση του δίσκου σε πραγματικό χρόνο
 - Αλλά η λογική που υπάρχει πάνω στο δίσκο την ξέρει



Πλάι – πλάι...

- LOOK εναντίον SCAN
 - Η SCAN διατρέχει το δίσκο από άκρη σε άκρη
 - Η LOOK πάει μόνο μέχρι την πιο απομακρυσμένη αίτηση
- Κυκλική διάτρεξη εναντίον διάτρεξης 2 κατευθύνσεων
 - 2 κατευθύνσεων: Αλλάζει κατεύθυνση διάτρεξης στα άκρα
 - Κυκλική: Όταν φτάσει στο τέλος επιστρέφει απευθείας στην αρχή και τότε ξεκινάει επόμενη διάτρεξη
 - Η 2 κατευθύνσεων είναι άδικη
 - Ευνοεί τους τομείς στο κέντρο του δίσκου
- Γήρανση αιτήσεων
 - Οι πολύ αποδοτικοί αλγόριθμοι μπορεί να οδηγήσουν σε λιμοκτονία
 - Γήρανση αιτήσεων: Εξασφαλίζει ότι παλιές αιτήσεις θα εξυπηρετηθούν κάποια στιγμή
 - Κοστίζει όμως λίγο σε επίδοση