ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ

1η σειρά Γραπτών Ασκήσεων

Όνομα: Στρατίγης Χαρίλαος ΑΜ: 03112136

Άσκηση 1

(α)

Γνωστή Ιεραρχία:

$$O(1) \subset O(\log^* n) \subset O(\log n) \subset O(\text{poly}(\log n)) \subset O(\sqrt{n}) \subset O(n)$$
$$\subset O(n \log n) \subset O(\text{poly}(n)) \subset O(n^{\log n}) \subset O(2^n) \subset O(3^n)$$
$$\subset O(n!) \subset O(n^n) \subset O(A(n))$$

- $n^2 = \boxed{O(n^2)}$
- $2^{(\log_2 n)^3} = \left(2^{\log_2 n}\right)^{\log^2 n} = n^{\log^2 n} = \boxed{O(n^{\log^2 n})}$
- $\frac{\log(n!)}{\log^3 n}$ (Είχαμε ότι $\log(n!) = \Theta(n \log n)$)

  Άρα $= \boxed{O\left(\frac{n}{\log^2 n}\right)}$
- $n \cdot 2^{2^{100}} = \boxed{O(n)}$
- $\log\left(\binom{n}{\log n}\right)$

Γνωρίζουμε ότι $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{n \cdot e}{k}\right)^k \xrightarrow{k=\log n} \left(\frac{n}{\log n}\right)^{\log n} \leq \binom{n}{\log n} \leq \left(\frac{e \cdot n}{\log n}\right)^{\log n} \xrightarrow{\log}$

$$\Rightarrow \log\left(\binom{n}{\log n}\right) \leq \log\left(\frac{e \cdot n}{\log n}\right)^{\log n} \Leftrightarrow \log\left(\binom{n}{\log n}\right) \leq \log\log\left(\frac{e \cdot n}{\log n}\right)$$

$\Leftrightarrow \log\left(\binom{n}{\log n}\right) \leq \log^2\left(\frac{en}{\log n}\right)$

Άρα $\log\left(\binom{n}{\log n}\right) = O(\log^2 n)$

- $\sum_{k=1}^{n} k^3 = \frac{1}{4} \cdot n^2 \cdot (n+1)^2 = \Theta(n^4)$
- $\log^4 n$ $\left(\forall \varepsilon > 0 : \log^d n = O(n^\varepsilon)\right.$ $\left.\text{(δώ για } \varepsilon = \frac{1}{2}\text{)}\right) = O(\sqrt{n})$
- $\sqrt{n!} = \Theta(\sqrt{n!})$ $\quad (f(n) = O(g(n)) \Rightarrow \sqrt{f(n)} = O(\sqrt{g(n)}))$
- $\binom{n}{6} = \frac{n!}{6!(n-6)!} = \frac{n \cdot (n-1)(n-2)\cdots 6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1}{1\cdot 2\cdot 3\cdot 4\cdot 5\cdot 6\,(n-6)(n-5)\cdots} = \cdots = \frac{n(n-1)(n-2)\cdots(n-5)}{1\cdot 2\cdot 3\cdot 4\cdot 5\cdot 6} = O(n^6)$
- $\frac{n^3}{\log^{10} n} \leq \frac{n^3}{\sqrt{n}} \quad (\log n \leq \sqrt{n}) = \Theta(n^{\frac{5}{2}})$
- $\left(\log_2(16n)\right)^{\log_2(16n)}$ $\left\{ a^{\log_b c} = c^{\log_b a} \right\} \Rightarrow 16n^{\log(\log(16n))} = \Theta(n^{\log\log n})$
- $\log\left(\binom{2n}{n}\right) \leq \log\left(\frac{2n^{2n}}{n^n (2n-n)^{2n-n}}\right) = \log 2^{2n} = \log 4^n = \Theta(n)$
- $n \sum_{k=0}^{n} \binom{n}{k} = O(n^{2n})$
- $\binom{2n}{n/4} \leq \frac{2n^{2n}}{(n/4)^{n/4} \cdot (2n - n/4)^{2n - n/4}} = \left(\frac{2^2}{(3/4)^{3/4} \cdot (1/4)^{1/4}}\right)^n = 7{,}1^n = O(7{,}1)^n$

$\left(\binom{n}{k} \leq \frac{n^n}{k^k (n-k)^{n-k}} \right)$

- $\sum_{k=1}^{n} k 2^k = \Theta(n 2^n)$
- $\sum_{k=1}^{n} k \cdot 2^{-k} = 2^{-n}(-n + 2^{n+1} - 2) = O(1)$

(B)

1) $T(n) = 3T(\frac{n}{4}) + n\log^5 n$

$n^{\log_4 a} = n^{\log_4 3} = n^{0.7} < n\log^5 n \Rightarrow$ Ση περ.

$af(\frac{n}{b}) < f(n)$ ισχύει για μεγάλο n ασυμπτωτικά

Άρα $T(n) = \Theta(n\log^5 n)$

2) $T(n) = 4T(\frac{n}{4}) + n\log^5 n$

$n^{\log_4 a} = n^{\log_4 4} = n$, όχι πολυωνυμική διαφορά με την $n\log^5 n$

Με δέντρο αναδρομών: Το δέντρο έχει ύψος $\log_4 n$ αφού κάθε υποπρόβλημα για κάθε κόμβο του δέντρου εξαφανίζεται σε βάθος i έχει μέγεθος $\frac{n}{4^i}$

Σε κάθε επίπεδο έχουμε 4 νέους κόμβους

Άρα $\sum_{i=0}^{\log_4 n-1} n\log^5 \frac{n}{4^i} = \Theta(n\log^6 n)$

3) $T(n) = 5T(\frac{n}{4}) + n\log^5 n$

$n^{\log_4 a} = n^{\log_4 5} = n^{1,16} < n\log^5 n \Rightarrow$ Ση περ.

Άρα $T(n) = \Theta(n^{\log_4 5})$

4) $T(n) = 2T(\frac{n}{3}) + \frac{n}{\log^5 n}$ $\quad n^{\log_3 a} = n^{\log_3 2} = n^{0,63} < \frac{n}{\log^5 n} \Rightarrow$ Ση περ.

$af(\frac{n}{b}) < f(n)$ ισχύει ασυμπτωτικά για μεγάλο n

Άρα $T(n) = \Theta(\frac{n}{\log^5 n})$

5) $T(n) = 2T\left(\frac{n}{7}\right) + 4T\left(\frac{n}{9}\right) + n$

Οχι Master Theorem

Βλέπουμε ότι στην αναδρομική σχέση φαίνεται να κυριαρχεί ο σταθερός όρος n όσο και αν διαλέξουμε "διαφορετικά".

Για να δείξουμε ότι ο αλγόριθμος είναι $O(n)$ αρκεί

νδ.ο $\exists\, c : T(k) < ck \quad \forall\, k < n$

$$T(n) < \frac{2cn}{7} + \frac{4cn}{9} + n = \left(1 + \frac{46c}{63}\right) n \leq cn \quad \text{για } c > \frac{63}{17}$$

Επίσης $T(n) = T\left(\frac{n}{6}\right) + 3T\left(\frac{n}{5}\right) + n \geq n$

Άρα $T(n) = \Theta(n)$

6) $T(n) = T(n-1) + \log n$

Δεν εφαρμόζεται Μ.Τ.

Πάμε με αναδρομή:

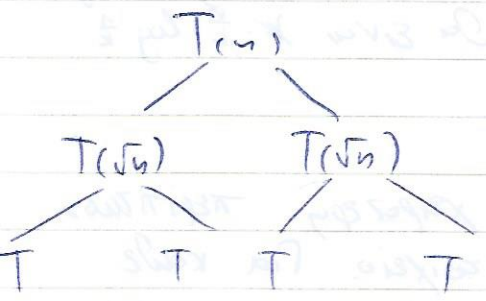
$$\left.\begin{aligned} T(n) &= T(n-1) + \log n \\ T(n-1) &= T(n-2) + \log(n-1) \\ &\vdots \\ T(1) &= T(0) + \log 1 \end{aligned}\right\} \overset{(+)}{\Longrightarrow} T(n) = T(0) + \log n + \log(n-1) + \ldots + \log 1$$

$$\Rightarrow T(n) = T(0) + \log(n(n-1)(n-2)\ldots 1)$$

$$\Rightarrow T(n) = T(0) + \log(n!) \quad \left(\log(n!) = O(n \log n)\right)$$

Άρα $T(n) = \Theta(n \log n)$

7) $T(n) = 2T(\sqrt{n}) + \Theta(\log n)$



Κοστος ανα επιπεδο = $\Theta(\log n)$
Συνολικα $\log(\log n)$ επιπεδα

Αρα $T(n) = \Theta(\log(\log(\log n)))$

8) $T(n) = T(3/4) + \sqrt{n}$

$n^{\log_b a} = n^{\log_4 1} = 1 < \sqrt{n} \Rightarrow$ 3η περ. Μ.Τ

α $f(\frac{n}{b}) \leq c \cdot f(n)$ για μεγαλο n

Αρα $T(n) = \Theta(\sqrt{n})$

Ασκηση 2

(α)

1) Εδω θα χρησιμοποιήσουμε την quickselect, η οποία έχει κόστος $O(n)$ και δίνοντάς της τον πίνακα A και μια επιλογή i. Θέση i θα επιστρέφει τον πίνακα μερικώς ταξινομημένο βάση της επιλογής θέσης. Όλα τα στοιχεία αριστερά από τη θέση i θα είναι μικρότερα του $A[i]$, και όλα που θα είναι δεξιά θα είναι μεγαλύτερα. Θα χρειαστεί να την καλέσουμε $\log k$ φορές*. Επομένως ο αλγόριθμος μας θα είναι $O(n \log k)$

*για το μεσαίο στοιχείο κάθε κομματιού

2) Το πρόβλημα αυτό είναι ισοδύναμο με την ταξινόμηση k πινάκων. Χρησιμοποιώντας quicksort, θα έχω κόστος $\frac{n}{k}\log\frac{n}{k}$ για κάθε πινακάκι. Αφού έχω k πίνακες το κόστος θα είναι $k \cdot \frac{n}{k} \cdot \log\frac{n}{k}$
$= O(n \log\frac{n}{k})$

Αφού κάθε στοιχείο έχει $\frac{n}{k}$ πιθανές θέσεις, στην χειρότερη περίπτωση θα χρειαστούμε $\log\frac{n}{k}$ συγκρίσεις για κάθε στοιχείο. Για κάθε πίνακα $\frac{n}{k}$ στοιχείων, λοιπόν, θα έχω στην χειρότερη περίπτωση κόστος $\frac{n}{k}\log\frac{n}{k}$, άρα $\Theta(n \log\frac{n}{k})$ για k πίνακες

(β)

1) Γνωρίζουμε ότι κάθε στοιχείο απέχει το πολύ k θέσεις από την επιθυμητή. Επομένως έχοντας εύρος $2k$, αν ταξινομήσουμε τα πρώτα $2k$ στοιχεία γνωρίζουμε ότι τα πρώτα k θα έχουν φτάσει στη σωστή τους θέση. Επαναλαμβάνουμε τη διαδικασία για $[k+1, 3k]$, $[2k+1, 4k]$ κ.ο.κ. Μέχρις ότου ταξινομηθούν όλα. Η διαδικασία αυτή θα γίνει $\frac{n}{2}$ φορές και έχει κόστος $2k\log 2k = O(k\log k)$
Επομένως ο αλγόριθμος θα είναι $O(\frac{n}{k} k\log k) = O(n\log k)$

2) Όπως και στην προηγούμενη άσκηση, για κάθε στοιχείο έχει $2k$ πιθανές θέσεις, στην χειρότερη περίπτωση θα χρειαστούν $\log 2k$ συγκρίσεις για κάθε στοιχείο. Επομένως για n στοιχεία θα χρειαστούν στην χειρότερη περίπτωση $n\log 2k$ στοιχεία $\rightarrow O(n\log k)$

Άσκηση 3

(α)

Για να λύσουμε το πρόβλημα θα χρησιμοποιήσουμε δυαδική κωδικοποίηση, δηλαδή τη δυνατότητα να αναπαραστήσουμε οποιονδήποτε αριθμό μεχρι το 1 εκατ. με μοναδικό τρόπο από 20 bits 0 και 1. Έτσι θα χρειαστούμε μόνο 20 ενδιάμεσους και θα αντιστοιχίσουμε κάθε ενδιάμεσο με ένα bit (με την προϋπόθεση ότι οι ενδιάμεσοι θα έχουν διαφορετικές θέσεις). Ο καθένας θα πήγαινε από το φίλτρο, μόνο εάν το φίλτρο έχει στη δυαδική του αναπαράσταση 1 στην αντίστοιχη θέση του ενδιάμεσου. Έτσι, αφού υπάρχει μόνο ένα μαγικό φίλτρο, ο συνδυασμός των ενδιάμεσων με υπερδυνάμεις θα υποδεικνύει την κωδικοποίηση του μαγικού φίλτρου.

(β) Σε αυτό το πρόβλημα θα χρησιμοποιήσουμε τη λογική της δυαδικής αναζήτησης και θα ακολουθήσουμε τον εξής συλλογισμό: θα δεσμεύσουμε υποδιαστήματα αριθμών για την αναζήτησή μας ως εξής: θα θεσμεύσουμε το μέσο d και ως άνω το άθροισμα όλων των d. Αρχικά θέτουμε ως $\lambda = \frac{min+max}{2}$ το άθροισμα των χλμ/ημέρα και αρχίζουμε να φτιάχνουμε πακέτα από d τέτοια ώστε να είναι μικρότερα του λ, μετρώντας πόσες μέρες θα χρειαστούμε (μ). Αν το μ είναι μικρότερο από το επιθυμητό κ, τότε επαναλαμβάνουμε τη διαδικασία αλλάζοντας το max σε λ, ενώ αν είναι μεγαλύτερο του κ, τότε αλλάζουμε το min σε λ. Η διαδικασία επαναλαμβάνεται εως ότου προκύψει $\mu = \kappa$. Κάθε επανάληψη έχει κόστος $\Theta(n)$ και έχουμε $\log(max-min) \leq \log \sum_{i=1}^{n} d_i$ επαναλήψεις.

Επομένως ο αλγόριθμος θα είναι πολυπλοκότητας $O(n \log(\sum_{i=1}^{n} d_i))$

Άσκηση 4

Σε αυτό το πρόβλημα θέλουμε να αποφανθούμε αν υπάρχει πλειοψηφία μεταξύ των βουλευτών, και αν ναι να πούμε ποιοι βουλευτές ανήκαν σε αυτή.

Αρχικά θα χωρίσουμε όλους τους βουλευτές σε ζευγάρια και θα τους βάλουμε να χαιρετηθούν. Αν ανήκουν στο ίδιο κόμμα, τότε θα κρατήσουμε τον ένα εξ αυτών. Αν δεν ανήκουν στο ίδιο κόμμα δεν θα κρατήσουμε κανέναν. Επίσης αν ο αριθμός των συνολικών βουλευτών είναι περιττός, τότε θα κρατήσουμε και τον τελευταίο που δεν είχε ζευγάρι.

Με αυτόν τον τρόπο είμαστε πάντα θα μας μένει ένας βουλευτής από το κόμμα της πλειοψηφίας (αν αυτό υπάρχει). Αυτό συμβαίνει γιατί αν κάποιος βουλευτής της πλειοψηφίας χαιρετηθεί με κάποιον από άλλο κόμμα, δεν θα τον κρατήσουμε, αλλά ταυτόχρονα θα "πετάξουμε" ένα από μη-πλειοψηφικό κόμμα. Έτσι θα διατηρηθεί η πλειοψηφία και στο επόμενο επίπεδο.

Επαναλαμβάνουμε τη διαδικασία μέχρις ότου μείνει μόνο ένας βουλευτής ο οποίος θα ανήκει στην πλειοψηφία (αν αυτή υπάρχει). Τότε, προκειμένου να μην τον βάλουμε να χαιρετηθεί ξανά με κάποιον, θα πάρουμε το ζευγάρι του (προ-τελευταίος βουλευτής που έμεινε) και θα τον βάλουμε να χαιρετηθεί με όλους από την αρχή ώστε να αναγνωρίσουμε και τους υπόλοιπους του κόμματος.

Εάν ο ζητούμενος βουλευτής ήταν αυτός που δεν είχε ζευγάρι, τότε βάζουμε αυτόν να χαιρετηθεί με όλους απ' την αρχή.

Σε κάθε επανάληψη θα χρειάζονται το πολύ αριθμός χειραψιών από την προηγούμενη φορά, δηλ. $n/2 + n/4 + \ldots \approx n$

Επίσης θα χρειάζονταν άλλες n χειραψίες για το τελικό πέρασμα.

Επομένως συνολικά $3n$ χειραψίες στη χειρότερη $\rightarrow O(n)$

(β) Εστω ότι βρισκόμαστε στην δεύτερη περίπτωση, δηλαδή υπάρχουν τουλάχιστον $n/3$ 1. Επομένως λάθος θα κάνουμε αν επιλέξουμε κ στοιχεία και είναι όλα μηδενικά.
Η πιθανότητα να επιλέξουμε 0 είναι $2/3$ (στην χειρότερη περίπτωση που υπάρχουν $2/3$ μηδενικά και μόνο $1/3$ άσσοι.)

Επομένως θέλουμε $\left(\frac{2}{3}\right)^{\kappa} < 0,1 \Rightarrow \kappa > 5,6$

* Αν αποφανθείτε ότι ο πίνακας μας έχει μόνο μηδενικά και είμαστε στην πρώτη περίπτωση, τότε θα μαντεύαμε σωστά. Επομένως δεν την συνυπολογίζουμε στην περίπτωση λάθους.

Συνεπώς με μόνο 6 τυχαίες επιλογές στοιχείων μπορούμε με πιθανότητα επιτυχίας >90% να το είδος του πίνακα
Ο χρόνος εκτέλεσης χειρότερης περίπτωσης για ντετερμινιστικό αλγόριθμο θα είναι $(2n/3 + 1)$ καθώς με τόσες δοκιμές θα είμαστε σίγουροι ότι δεν θα υπάρχει κανένας άσσος (αν δεν υπάρχει)

Άσκηση 5

Για να λύσουμε το πρόβλημα ξεκινάμε από το δεξιότερο κτίριο. Η λογική που θα ακολουθήσουμε είναι ότι θα συγκρίνουμε κάθε κτίριο με το αμέσως αριστερά του. Αν το αριστερά κτίριο είναι ψηλότερο, τότε συμπληρώνουμε τον πίνακα Β. Εαν όχι, τότε κρατάμε το τρέχον κτίριο σε μια στοίβα προκειμένου να το συμπληρώσουμε αργότερα και προχωράμε στο επόμενο κτίριο ακολουθώντας την ίδια διαδικασία. Όταν βρεθεί ένα κτίριο ψηλότερο από το προηγούμενό του, τότε εκτός από την αρχική σύγκριση που κάναμε, το συγκρίναμε και με τα στοιχεία της στοίβας μας, αδειάζοντας την από κάθε κτίριο που είναι πιο κοντό από αυτό που βρήκαμε.
Προφανώς στο Β[ε] θα υπάρχει η τιμή 0.
Με αυτον τον τροπο δεν χρειάζονται η συγκρίσεις για κάθε κτίριο αλλά κ(περπατημένο). Συνεπώς ο αλγόριθμος μας είναι γραμμικος και όχι πολυπλοκότητας $O(n^2)$