ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ - ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
Σχεδίαση και Ανάλυση Αλγορίθμων
Διδάσκων: Ε. Μαρκάκης, Φθινοπωρινό εξάμηνο 2014-2015

1η Σειρά Ασκήσεων **Εξέταση: 4 Νοεμβρίου 2014**

Πρόβλημα 1. Απαντήστε στα παρακάτω 2 ερωτήματα, σχετικά με την ασυμπτωτική συμπεριφορά συναρτήσεων.

(i) Έστω $c \in \mathbb{R}^+$, και $g = 1 + c + c^2 + \cdots + c^n$. Να δείξετε ότι

- $g \in \Theta(1)$, αν $c < 1$
- $g \in \Theta(n)$, αν $c = 1$
- $g \in \Theta(c^n)$, αν $c > 1$

(ii) Να διατάξετε τις ακόλουθες συναρτήσεις με βάση τον αυξητικό τους χαρακτήρα. Ενδεχομένως κάποιες από τις συναρτήσεις αυτές να έχουν την ίδια ασυμπτωτική συμπεριφορά και να υπάρχουν παραπάνω από μια σωστές διατάξεις.

$$\log\log n^4,\ \sqrt{n},\ \log_4 n^4,\ n^{1/\sqrt{n}},\ \sqrt{4^{logn}},\ n^4\log n,\ n^n,\ n^4,\ 2^{n\log n},\ \log 3^n \log n$$

Πρόβλημα 2. Θέλετε να επιλέξετε για κάποιο πρόβλημα έναν από τους παρακάτω τρεις αλγορίθμους:

1. Ο αλγόριθμος A επιλύει το αρχικό πρόβλημα με διαίρεσή του σε 5 υποπροβλήματα του μισού μεγέθους, αναδρομική επίλυση των υποπροβλημάτων, και μετά συνδυασμό των λύσεων σε γραμικό χρόνο.

2. Ο αλγόριθμος B επιλύει προβλήματα μεγέθους n με αναδρομική επίλυση 2 υποπροβλημάτων μεγέθους $n-1$, και μετά συνδυασμό των λύσεων σε σταθερό χρόνο.

3. Ο αλγόριθμος Γ επιλύει το αρχικό πρόβλημα με διαίρεσή του σε 9 υποπροβλήματα με μέγεθος το ένα τρίτο του αρχικού, αναδρομική επίλυση των υποπροβλημάτων, και μετά συνδυασμό των λύσεων σε τετραγωνικό χρόνο.

Ποιοι είναι οι χρόνοι εκτέλεσης των αλγορίθμων και ποιον θα επιλέγατε από τους τρεις; Ποιος είναι ο χειρότερος αλγόριθμος από τους τρεις;

Πρόβλημα 3. Έστω ότι θέλετε να ταξινομήσετε n αριθμούς, οι οποίοι βρίσκονται αποθηκευμένοι σε έναν πίνακα. Είδαμε στο μάθημα ότι μπορούμε να λύσουμε το πρόβλημα σε χρόνο $O(nlogn)$. Έστω τώρα ότι έχουμε την πληροφορία ότι οι αριθμοί αυτοί είναι θετικοί ακέραιοι με τιμές από το $\{1, 2, ..., M\}$. Σχεδιάστε έναν αλγόριθμο ο οποίος ταξινομεί τους αριθμούς σε χρόνο $O(n + M)$. Αυτό σημαίνει ότι όταν το M είναι μια μικρή σταθερά, έχουμε καλύτερο χρόνο από $O(nlogn)$.

Πρόβλημα 4. Έστω ότι έχετε έναν ταξινομημένο πίνακα διαφορετικών μεταξύ τους ακεραίων $A[1..n]$. Σχεδιάστε έναν αλγόριθμο που αποφασίζει αν υπάρχει δείκτης i, για τον οποίο $A[i] = i$. Ο αλγόριθμος θα πρέπει να εκτελείται σε χρόνο $O(logn)$.

Πρόβλημα 5. Δύο πίνακες ακεραίων είναι αποθηκευμένοι σε δύο ξεχωριστές βάσεις δεδομένων. Κάθε πίνακας περιέχει n στοιχεία και υποθέστε ότι οι $2n$ ακέραιοι είναι όλοι διαφορετικοί μεταξύ τους. Θέλετε να υπολογίσετε τη διάμεσο των στοιχείων αυτών, δηλαδή το n-οστό μικρότερο στοιχείο. Έχουμε δει πώς να βρεθεί η διάμεσος στοιχείων σε χρόνο $O(n)$, αν είχαμε πρόσβαση σε όλα τα στοιχεία και τα αποθηκεύαμε όλα μαζί σε έναν πίνακα. Η πρόσβαση στις βάσεις αυτές όμως έχει κάποιο κόστος και ο μόνος τρόπος πρόσβασης είναι με τον εξής τύπο από queries: Μπορείτε να δίνετε μια παράμετρο k σε οποιαδήποτε από τις 2 βάσεις, και να σας επιστρέφεται το k-οστό μικρότερο στοιχείο από τη συγκεκριμένη βάση. Σχεδιάστε έναν αλγόριθμο που βρίσκει τη διάμεσο με $O(logn)$ queries.

Πρόβλημα 6. Σας έχουν προσλάβει ως σύμβουλο σε μια εταιρεία με φορτηγά που εκτελούν καθημερινά το δρομολόγιο Αθήνα - Θεσσαλονίκη, μεταφέροντας δέματα, κούτες, και άλλα πράγματα. Κάθε μέρα, η εταιρεία εφαρμόζει τον ίδιο αλγόριθμο προκειμένου να ελαχιστοποιήσει τον αριθμό των φορτηγών που θα αναχωρήσουν από Αθήνα για Θεσσαλονίκη. Η επεξεργασία των αιτημάτων στην Αθήνα γίνεται με τη σειρά που έρχονται (για να μην υπάρχουν παράπονα από τους πελάτες). Ας υποθέσουμε ότι κάθε αίτημα αντιστοιχεί στην αποστολή μιας κούτας, και το i-οστό αίτημα έχει βάρος w_i. Τα φορτηγά έχουν όλα το ίδιο άνω όριο W για το βάρος που μπορούν να μεταφέρουν. Βλέποντας τα αιτήματα, οι υπάλληλοι της εταιρείας αρχίζουν και φορτώνουν κούτες στο πρώτο φορτηγό μέχρι το σημείο όπου το φορτηγό δεν μπορεί να δεχτεί άλλη κούτα λόγω του περιορισμού βάρους. Στη συνέχεια κάνουν το ίδιο με το επόμενο φορτηγό μέχρι να τελειώσουν τα αιτήματα. Π.χ., αν τα αιτήματα με τη σειρά που έρχονται είχαν βάρη $(10, 7, 2, 3, 11, 4)$, και $W = 20$, τότε τα πρώτα 3 αιτήματα θα πάνε στο φορτηγό 1, και τα επόμενα 3 στο φορτηγό 2. Έτσι θα είχαμε 2 φορτηγά εκείνη τη μέρα που θα έκαναν το δρομολόγιο Αθήνα-Θεσσαλονίκη.

Η εταιρεία ανησυχεί ότι με τον αλγόριθμο που εφαρμόζει ενδεχομένως να χρησιμοποιεί περισσότερα φορτηγά από όσα χρειάζονται. Για το λόγο αυτό σας έχει καλέσει να βοηθήσετε. Σκοπός της άσκησης είναι να αποδείξετε ότι για οποιαδήποτε ακολουθία από αιτήματα με βάρη $(w_1, ..., w_n)$, όπου w_i είναι το βάρος του i-οστού αιτήματος, ο αλγόριθμος που εφαρμόζει η εταιρεία είναι βέλτιστος, δηλαδή ελαχιστοποιεί τον αριθμό των φορτηγών που χρειάζονται για να ικανοποιηθούν όλα τα αιτήματα.