

Άσκηση 4

Ημερομηνία Παράδοσης: 23 Μαΐου 2008

Σημειώσεις:

1. Στις απαντήσεις που θα παραδώσετε σημειώστε στην πρώτη σελίδα το ονοματεπώνυμό σας, τον αριθμό μητρώου σας και το τμήμα σας.
2. Οι ασκήσεις πρέπει να γίνουν ατομικά. Οποιαδήποτε μορφή αντιγραφής απαγορεύεται.
3. Η παρούσα άσκηση πρέπει να παραδοθεί το αργότερο μέχρι την αρχή του μαθήματος της 23ης Μαΐου, δηλαδή μέχρι τις 11:15. Καθυστερημένες ασκήσεις δε θα γίνουν δεκτές.
4. Σε περίπτωση που έχετε ερωτήσεις στείλτε email στην ηλεκτρονική λίστα του μαθήματος: hy240-list@tem.uoc.gr

Πρόβλημα 1 [10 μονάδες] Περιγράψτε την εισαγωγή των κλειδιών 5, 28, 19, 15, 20, 33, 12, 17 (με αυτή τη σειρά) σε πίνακα κατακερματισμού όπου οι συγκρούσεις αντιμετωπίζονται με linear probing. Υποθέστε ότι ο πίνακας έχει 9 θέσεις και ότι η συνάρτηση κατακερματισμού είναι $h(k) = k \bmod 9$. Σχεδιάστε την κατάσταση του πίνακα κατακερματισμού μετά από κάθε εισαγωγή.

Πρόβλημα 2 [30 μονάδες] Θεωρήστε πίνακα κατακερματισμού μεγέθους $m = 11$ και δύο συναρτήσεις κατακερματισμού h_1 και h_2 :

$$h_1(x) = ((\text{άθροισμα τιμών πρώτου και τελευταίου γράμματος της})x) \bmod m$$

$$h_2(x) = ((\text{τιμή τελευταίου γράμματος}) \bmod (m - 1)) + 1$$

όπου η τιμή (value) ενός γράμματος είναι η θέση του στο αλφάβητο (για παράδειγμα, $\text{value}(a)=1$, $\text{value}(b)=2$, κλπ.). Στην άσκηση αυτή θα αναφερόμαστε σε λέξεις του αγγλικού αλφαβήτου. Κάποια παράδειγμα τιμών της συνάρτησης κατακερματισμού δίνονται στον παρακάτω πίνακα.

x	$h_1(x)$	$h_2(x)$
get	5	1
word	5	5
line	6	6
help	2	7
right	5	1
cancel	4	3
compiler	10	9

1. Σχεδιάστε εικόνα του πίνακα κατακερματισμού μετά την εισαγωγή κάθε μίας εκ των παρακάτω λέξεων (οι λέξεις εισάγονται με τη σειρά που σας δίνονται):

ibex, hare, ape, bat, koala, mud, dog, carp, stork

2. Σημειώστε τα κελιά του πίνακα κατακερματισμού που επισκεπτόμαστε όταν αναζητούμε τη λέξη bird.

Κάντε τα παραπάνω δύο ερωτήματα για κάθε μία από τις εξής τεχνικές κατακερματισμού:

- chaining με την h_1 ως συνάρτηση κατακερματισμού
- linear probing με την h_1 ως συνάρτηση κατακερματισμού
- double hashing με την h_1 ως πρωτεύουσα συνάρτηση κατακερματισμού και με την h_2 ως δευτερεύουσα συνάρτηση κατακερματισμού.

Πρόβλημα 3 [20 μονάδες] Υπάρχει ένα πολύ απλός αλγόριθμος ταξινόμησης που ονομάζεται CountSort ο οποίος θα κατασκευάσει μία νέα ταξινομημένη λίστα από μία λίστα L , υπό την προϋπόθεση ότι όλα τα κλειδιά στην L είναι ανά δύο διαφορετικά. Ο CountSort διαβάζει την L μία φορά και για κάθε κλειδί k στην L διατρέχει την L για να δει πόσα κλειδιά είναι μικρότερα από το k . Αν c είναι το πλήθος των μικρότερων από το k κλειδιών, τότε η σωστή θέση του k στην ταξινομημένη λίστα είναι η θέση $c + 1$.

1. Γράψτε ψευδοκώδικα για τον αλγόριθμο CountSort.
2. Υπολογίστε τον αριθμό των συγκρίσεων που εκτελεί ο CountSort.

Πρόβλημα 4 [20 μονάδες] Σας δίδεται σύνολο S από n πραγματικούς αριθμούς και ένας επιπλέον πραγματικός αριθμός x . Περιγράψτε έναν $O(n \log n)$ αλγόριθμο ο οποίος καθορίζει αν υπάρχουν δύο στοιχεία στο σύνολο S των οποίων το άθροισμα είναι ακριβώς x .

Υπόδειξη: Χρησιμοποιήστε ταξινόμηση.

Πρόβλημα 5 [20 μονάδες] Θεωρήστε τη συνάρτηση foo που σας δίδεται παρακάτω (χωρίς σχόλια):

```
foo(int a[], int l, int r, int k)
{ int i;
  i = partition(a,l,r);
  if (i == k) return a[k];
  else if (i < k) return foo(a,i+1,r,k);
  else return foo(a,l,i-1,k);
}
```

καθώς και τη συνάρτηση partition (με σχόλια):

```
partition(int a[], int l, int r)
/* let v=a[r]; this function rearranges the subarray a[l] through */
/* a[r] such that the elements less than or equal to v precede the */
/* remaining elements, and returns index i such that a[i]=v */
{
    int v, t, i, j;
    v = a[r]; i = l - 1; j = r;
    while (1) {
        while (a[++i] < v) ;
        while (a[--j] > v && j >= l) ;
        if (i >= j) break;
        t = a[i]; a[i] = a[j]; a[j] = t;
    }
    t = a[i]; a[i] = a[r]; a[r] = t;
    return i;
}
```

Έστω ότι ο a είναι πίνακας ακεραίων μεγέθους N , και $1 \leq k \leq N$.

1. Τί θα επιστρέψει η $\text{foo}(a, 1, N, k)$;
2. Ποιά είναι πολυπλοκότητα της $\text{foo}(a, 1, N, k)$ στη χειρότερη περίπτωση (συναρτήσει των N και k); Πότε (για τί δεδομένα δηλαδή) αυτή συμβαίνει;

Σύνολο μονάδων: 100