



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

## Τμήμα Πληροφορικής

ΕΠΑ 035- ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΗΛΕΚΤΡΟΛΟΓΟΥΣ  
ΜΗΧΑΝΙΚΟΥΣ ΚΑΙ ΜΗΧΑΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΤΩΝ

### Φροντιστηριακές Ασκήσεις - Κατακερματισμός

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

#### Άσκηση 1

Έστω το σύνολο  $\{9, 21, 8, 25, 26, 12, 13\}$ . Οι αριθμοί του συνόλου έχουν εισαχθεί σε ένα πίνακα κατακερματισμού μεγέθους 13 χρησιμοποιώντας τη συνάρτηση κατακερματισμού  $h(x) = x \bmod 13$ , και αντιμετωπίζοντας οποιεσδήποτε συγκρούσεις με τη μέθοδο της γραμμικής αναζήτησης ανοικτής διεύθυνσης. Το αποτέλεσμα των εισαγωγών, των οποίων η σειρά δεν μας είναι γνωστή, φαίνεται πιο κάτω

0	1	2	3	4	5	6	7	8	9	10	11	12
13	26	25						8	21	9		12

Θεωρήστε τις πιο κάτω προτάσεις:

- (α) Ο αριθμός 25 εισήχθηκε τελευταίος.
- (β) Τουλάχιστον τρεις αριθμοί εισήχθηκαν πριν από τον αριθμό 25.
- (γ) Ο αριθμός 8 εισήχθηκε πριν από τον αριθμό 12.
- (δ) Ο αριθμός 9 εισήχθηκε πριν από τον αριθμό 8.

Για κάθε μια από τις πιο πάνω προτάσεις, να αποφασίσετε κατά πόσον ισχύει

- (i) σε οποιαδήποτε σειρά εισαγωγής των στοιχείων που δημιουργεί τον πιο πάνω πίνακα.
- (ii) σε καμία σειρά εισαγωγής των στοιχείων που δημιουργεί τον πιο πάνω πίνακα
- (iii) σε κάποιες αλλά όχι όλες τις σειρές εισαγωγής των στοιχείων που δημιουργούν τον πιο πάνω πίνακα.

Να αιτιολογήσετε τις απαντήσεις σας.

#### Άσκηση 2

α) Να δείξετε το αποτέλεσμα της διαδοχικής εισαγωγής των κλειδιών  $\{5, 28, 19, 15, 20, 33, 12, 17, 29\}$  σε ένα πίνακα κατακερματισμού μεγέθους 9 με συνάρτηση κατακερματισμού  $h(k) = k \bmod 9$ , χρησιμοποιώντας τις ακόλουθες μεθόδους διαχείριση συγκρούσεων:

1) Αλυσίδωση (Chaining)

2) Γραμμική Αναζήτηση ανοικτής διεύθυνσης (Linear Probing)

3) Δευτεροβάθμια αναζήτηση ανοικτής διεύθυνσης (Quadratic Probing), όπου το μέγεθος του πίνακα είναι 11 και η αρχική συνάρτηση κατακερματισμού  $h(k)=k \bmod 11$ .

β) Τι εναλλακτικές δομές (αντί για συνδεδεμένες λίστες) θα μπορούσατε να χρησιμοποιήσετε στην τεχνική κατακερματισμού με αλυσίδωση. Ποια είναι τα πλεονεκτήματα και μειονεκτήματα της εναλλακτικής σας λύσης. Εκφράσετε τον χρόνο εκτέλεσης των διαδικασιών εισαγωγής και αναζήτησης συνάρτηση του load factor  $\lambda$  (όπου χρειάζεται)



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

## Τμήμα Πληροφορικής

ΕΠΑ 035- ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΗΛΕΚΤΡΟΛΟΓΟΥΣ  
ΜΗΧΑΝΙΚΟΥΣ ΚΑΙ ΜΗΧΑΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΤΩΝ

Φροντιστηριακές Ασκήσεις - Κατακερματισμός

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

## ΛΥΣΕΙΣ

### Λύση 1

Κατ' αρχήν σημειώνουμε το υπόλοιπο κάθε διαίρεσης

$$9 \quad \% \quad 13 = 9$$

$$21 \quad \% \quad 13 = 8$$

$$8 \quad \% \quad 13 = 8$$

$$25 \quad \% \quad 13 = 12$$

$$26 \quad \% \quad 13 = 0$$

$$12 \quad \% \quad 13 = 12$$

$$13 \quad \% \quad 13 = 0$$

Μετά βρίσκουμε ποια στοιχεία βρίσκονται ήδη στο σωστό bucket

Αυτά τα στοιχεία είναι το : 8, 12, 13.

Μετά βρίσκουμε ποια εξαρτώνται από αυτά τα στοιχεία. Καταλήγουμε στα εξής συμπεράσματα (σχέσεις) (όπου  $x \Rightarrow y$  δηλώνει ότι ο  $x$  εισήχθη πριν το  $y$ )

**A) 8  $\Rightarrow$  21  $\Rightarrow$  9**

**B) 12  $\Rightarrow$  13  $\Rightarrow$  26  $\Rightarrow$  25** (Εφόσον ο 26 αναγκάζεται να δημιουργήσει τέσσερα collision προτού εισαχθεί)

a) Ισχύει το (iii) – Δηλαδή ο 25 είναι τελευταίος σε κάποιες σειρές εισαγωγής.

25 τελευταίος : { 8,21,9, 12,13,26,25 }

25 μη-τελευταίος: { 12,13,26,25, 8,21,9 }

b) Ισχύει το (i) – Δηλαδή πάντα θα έχουν εισαχθεί τουλάχιστο 3 στοιχεία πριν το 25.

Εξάγεται κατευθείαν από την σχέση:

**13  $\Rightarrow$  26  $\Rightarrow$  25**

12  $\Rightarrow$  25

c) Ισχύει το (iii) – δηλαδή ο 8 μπορεί να εισήχθη πριν το 12 ή μετά το 12.

8 πριν 12 : { **8**,21,9, **12**,13,26,25 }

8 μετά 12 : { **12**,13,26,25, **8**,21,9 }

d) Ισχύει το (ii) – σε καμία σειρά εισαγωγής δεν μπορεί ο 9 να εισήχθη πριν το 8.

Εξάγεται κατευθείαν από την σχέση:

**A) 8  $\Rightarrow$  21  $\Rightarrow$  9**

## Λύση 2

a)  $5\%9=5$ ,  $28\%9=1$ ,  $19\%9=1$ ,  $15\%9=6$ ,  $20\%9=2$ ,  $33\%9=6$   
 $12\%9=3$ ,  $17\%9=8$ ,  $29\%9=2$

### 1) Αλυσίδωση (Chaining)

0 | x  
1 | 19, 28, x  
2 | 29, 20, x  
3 | 12, x  
4 | x  
5 | 5, x  
6 | 33, 15, x  
7 | x  
8 | 17, x

### 2) Γραμμική Αναζήτηση ανοικτής διεύθυνσης (Linear Probing)

Σημείωση: (cl:) = collision

$5\%9=5$ ,  $28\%9=1$ ,  $19\%9=1$ ,  $15\%9=6$ ,  $20\%9=2$ ,  $33\%9=6$   
 $12\%9=3$ ,  $17\%9=8$ ,  $29\%9=2$

0 | 29  
1 | 28 (cl:19)  
2 | 19 (cl:20,cl:29)  
3 | 20 (cl:12,cl:29)  
4 | 12 (cl:29)  
5 | 5 (cl:29)  
6 | 15 (cl:33,cl:29)  
7 | 33 (cl:29)  
8 | 17 (cl:29)

### 3) Δευτεροβάθμια αναζήτηση ανοικτής διεύθυνσης (Quadratic Probing)

$5\%11=5$ ,  $28\%11=6$ ,  $19\%11=8$ ,  $15\%11=4$ ,  $20\%11=9$ ,  $33\%11=0$   
 $12\%11=1$ ,  $17\%11=6$ ,  $29\%11=7$

0 | 33 ([X+4] cl:29)  
1 | 12 ([X+16] cl:29)  
2 |  
3 |  
4 | 15  
5 | 5 ([X+9] cl:29)  
6 | 28 (cl:17)  
7 | 17 ([X] cl:29)  
8 | 19 ([X+1] cl:29)  
9 | 20  
10 | 29 [X+25]

## b)

- **m-ary tree** : Εισαγωγή= $O(1+\log_m \lambda)$  Αναζήτηση= $O(1+\log_m \lambda)$

Μειονέκτημα: Η χρήση δυο pointers ανά κόμβο στοιχίζει ακόμη περισσότερη μνήμη από την μονά-συνδεδεμένη υλοποίηση της αλυσίδωσης.

- **Sorted Array (using realloc)**: Εισαγωγή= $O(1+\log_2 \lambda)$  Αναζήτηση= $O(1+\log_2 \lambda)$

(Αυτό μπορεί να υλοποιηθεί με την χρήση binary search)

Μειονέκτημα: Η χρήση realloc μπορεί να στοιχίσει  $O(1+\lambda)$  κόστος αν το λειτουργικό σύστημα δεν μπορεί να μας δώσει συνεχόμενα blocks μνήμης και αναγκαστεί να αντιγράψει τα στοιχεία σε νέα τοποθεσία μνήμης.