

1η Σειρά Ασκήσεων

Άσκηση 1

1α)Έχουμε 4 πλακέτες διπλής όψης δηλαδή 8 επιφάνιες. 4096 ίχνη ανα επιφάνεια οπότε 8×4096 .Έχουμε 1024 τομείς ανα ίχνος οπότε $8 \times 4096 \times 1024$.Τέλος έχουμε 512 bytes ανα τομέα οπότε προκύπτει $8 \times 4096 \times 1024 \times 512 = 17.179.869.184$ συνολικά bytes. Κάθε εγγραφή έχει χώρο για 256 bytes οπότε $17.179.869.184 / 256 = 67.108.864$ εγγράφες μπορούν να αποθηκεύτουν.

1β) $1.000.000 \times 256 = 256.000.000$ bytes
 $4096 / 256 = 16$ εγγραφες ανα μπλοκ
 $1.000.000 / 16 = 62.500$ μπλοκ

Για τυχαία προσπέλαση:

Average Seek Time =10ms

Avg Rotation delay= $\frac{1}{2} \times 60 / 7200 = 4,17$

ο χρόνος μετακίνησης στο επόμενο ίχνος είναι μηδαμινός οπότε έχουμε:

Χρόνος για τυχαία προσπέλαση= $X \times$ Average Seek Time + Average Rotation Delay.Οπότε περίπου $X \times 14,17$ ms

Για σειριακή προσπέλαση:

Μια ολόκληρη περιστροφή χρειάζεται $60 / 7200 = 0.00833 = 8,33$ ms

Σε κάθε περιστροφή διαβάζεται ένα ίχνος και κάθε ίχνος είναι 512 bytes άρα $512 \text{ bytes} / 8,33 \text{ ms} =$ περίπου 61,5kb/ms

Συνολικό μέγεθος = 256.000.000 bytes

οπότε $256.000.000 \text{ bytes} / 61,5 \text{ kb/ms} = 250.000 \text{ kb} / 61,5 \text{ kb/ms} =$ περίπου 4056ms

Αρα πρέπει να βρούμε X τέτοιο ώστε Χρόνος σειριακής <Χρόνος τυχαίας $4056 < X \Leftrightarrow 4056 / 14 < X =$ περίπου 289.

Άσκηση 2

2α)Κάθε εγγραφή είναι $6+25+25+8=64$ bytes . Συνολικές πλειάδες είναι 100.000 οπότε $100.000 \times 64 = 6.400.000$ bytes για όλες τις πλειάδες. Τέλος κάθε μπλοκ χωράει 512 bytes οπότε $6.400.000 / 512 = 12.500$ μπλοκ για όλο το αρχείο με τις πλειάδες.

2β)Tree pointer=4 bytes, Data pointer=5 bytes, D=8 bytes

Για τους κόμβους φύλλων έχουμε 8(μεγεθος d)+5(μέγεθος data pointer)= 13 bytes. Κάθε μπλοκ έχει χωρητικότητα 512 Bytes οπότε $512/13=39$ εγγραφές φύλλων. Για 100.000 χρειαζόμαστε $100.000/39=2.565$ φύλλα.

Το μέγεθος εσωτερικού κόμβου προκύπτει $D+ \text{Tree pointer} \Leftrightarrow 4+8=12$. Πάλι βρίσκουμε την χωρητικότητα ανα μπλοκ $512/12=$ περίπου 43 εγγραφές για εσωτερικούς κόμβους. Για τον ελάχιστο αριθμό μπλοκ θα φτιάξουμε το δέντρο από κάτω προς τα πάνω. Αρχικά έχουμε 2565 φύλλα. Μετά κάθε εσωτερικός κόμβος επιπέδου 2 μπορεί να έχει εως και 43 παιδία οπότε $2565/43=60$. Στο επίπεδο 3 μπορεί να έχει $60/43=2$ κόμβους. Τέλος έχουμε 1 ρίζα. Για τον ελάχιστο αριθμό μπλοκ έχουμε 2565 φύλλα+60 κόμβους επιπέδου 2 +2 κομβους επιπέδου 3+1 ρίζα= $2565+60+2+1=2628$ μπλοκ.

Όσον αφορά τον μέγιστο αριθμό μπλοκ γνωρίζουμε ότι η πληρότητα κάθε μπλοκ πρέπει να είναι τουλάχιστον 50%. συγκεκριμένα $(N+1)/2$ tree και data pointers . Οπότε $(39+1)/2=20$ εγγραφές για data pointers . Έτσι έχουμε μέγιστο αριθμό φύλλων $100.000/20=5000$. Για εσωτερικούς κόμβους έχουμε $(43+1)/2=22$ εγγραφές για tree pointers. Άρα $5000/22=228$ για επίπεδο 2, $228/22=11$ για επίπεδο 3 , 1 για ρίζα. Άρα 5000 για τα φύλλα $+228+11+1=5240$ μεγιστα μπλοκ Ελάχιστα μπλοκ= 2628
Μέγιστα μπλοκ= 5240

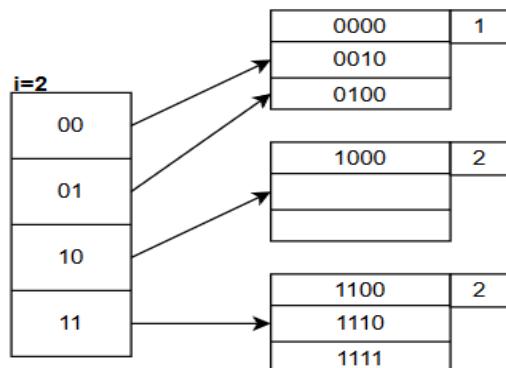
2γ)Αρχικά για τις 1000 εγγραφές έχουμε 39 διαθέσιμες για τα φύλλα εφόσον χρησιμοποιούμε τον μικρό δέντρο, οπότε $1000/39=26$ κόμβους φύλλων. Επιπλέον το μέγεθος κάθε εγγραφής είναι 64 bytes και το μέγεθος κάθε μπλοκ είναι 512 bytes. Άρα, κάθε μπλοκ δεδομένων χωράει $512/64=8$ εγγραφές. Συνεπώς, οι 1000 εγγραφές καταλαμβάνουν $1000/8=125$ μπλοκ. Για να προσπελάσουμε αυτά τα φύλλα, πρέπει να διαβαστεί πρώτα η διαδρομή από τη ρίζα μέχρι το πρώτο φύλλο, δηλαδή 1 μπλοκ από κάθε ενδιάμεσο επίπεδο του B+ δέντρου. Γνωρίζουμε ότι το δέντρο 4 επίπεδα (ρίζα ,2 ενδιάμεσα επίπεδα ,φύλλα), άρα διαβάζουμε 1 μπλοκ ρίζας, 1 μπλοκ 2ου επίπεδου, 1 μπλοκ 3ο επιπέδου, 26 φύλλα που αντιστοιχούν σε ενα μπλοκ το καθένα και 125 μπλοκ δεδομένων. Συνολικά προκύπτει $125+1+1+1+26=154$ μπλοκ πρέπει να διαβαστούν.

Άσκηση 3

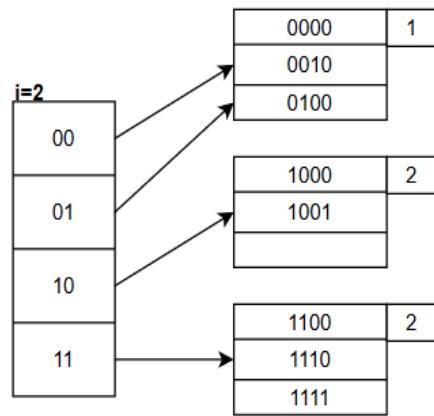
3α) Έχουμε 3 κλειδιά ανα κάδο. Το ολικό βάθος είναι 2 οπότε το ευρετήριο έχει 4 αρχικές εγγραφές που μπορεί να αντιστοιχεί μέχρι ενα bucket στο καθένα. Κάθε bucket έχει χώρο για 3 κλειδιά οπότε $4 \times 3 = 12$ είναι ο μέγιστος αριθμός τιμών που μπορούμε να έχουμε πριν αυξηθεί το ολικό βάθος. Οποιαδήποτε άλλη προσθήκη θα οδηγήσει σε υπερχείλιση ενός bucket με αποτέλεσμα να χρειαστεί να αυξηθεί το ολικό βάθος.

3β) Στο ευρετήριο υπάρχουν ήδη μέσα 6 τιμές . 3 στο πρώτο bucket , 1 στο δεύτερο, 2 στο τρίτο. Αν προσθέσουμε άλλη μια τιμή στο τελευταίο bucket θα γεμίσει και στην επόμενη προσθήκη τιμής ο κάδος θα υπερχειλίσει . Επειδή στον συγκεκριμένο bucket το local depth=global depth καί υπάρχει υπερχείλιση τότε θα πρέπει να διπλασιάσουμε το ευρετήριο. Άρα οι ελάχιστες τιμές που χρειάζονται για να διπλασιαστεί το ευρετήριο είναι 8. Οποιαδήποτε άλλος συνδυασμός από προσθήκες ξεπερνάει τις 8 τιμές

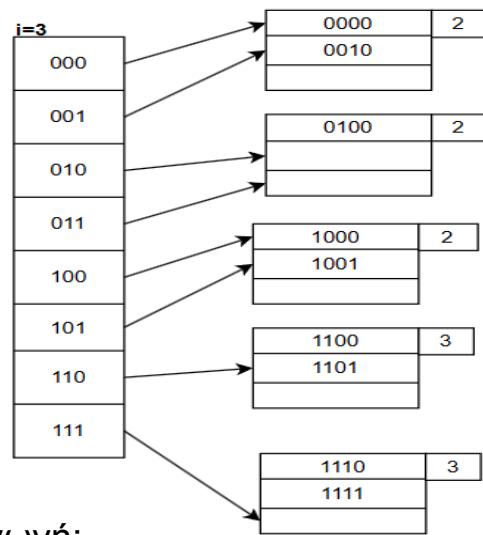
3γ) Μετά την πρώτη εισαγωγή:



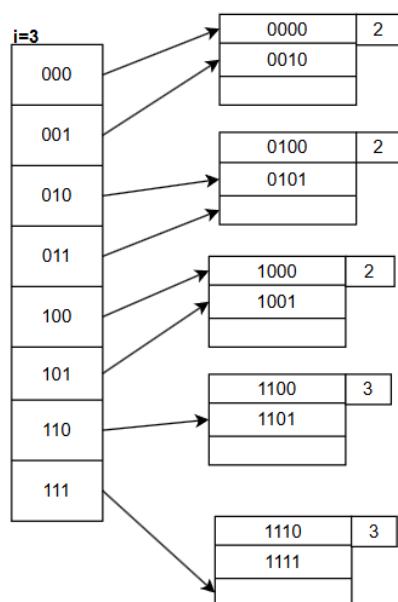
Μετά την δεύτερη εισαγωγή:
(συνέχεια στην επόμενη σελίδα)



Μετά την τρίτη εισαγωγή:



Τελευταία εισαγωγή:



Άσκηση 4

4α) $i=1, m=^*1$

4β)

Μετά την πρώτη προσθήκη το $U > 80$ οπότε κάνουμε split:

U=0,875

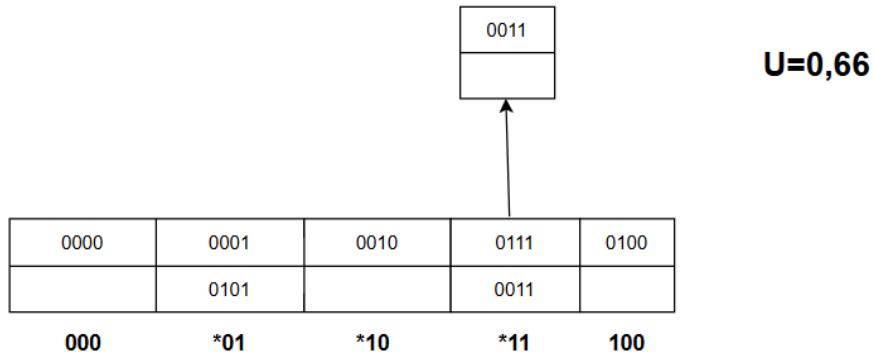
0100	0101	00010	0111
0000	0001		0011
00	01	10	11

Μετά το split το ευρετήριο είναι σε αυτή την μορφή:

U=0,7

0000	0001	0010	0111	0100
	0101		0011	
000	*01	*10	*11	100

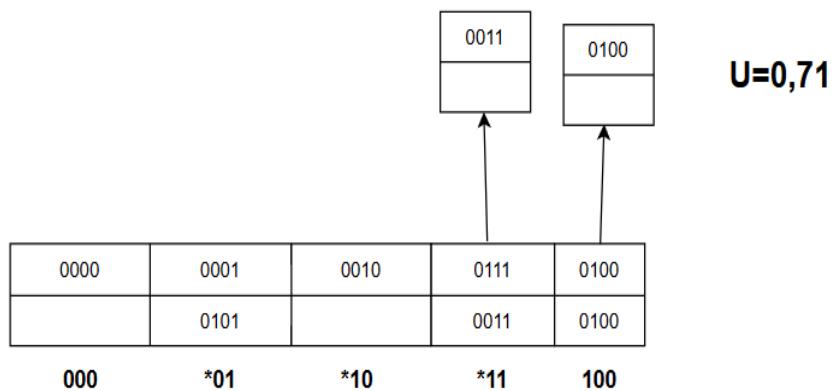
Με την 2η προσθήκη:



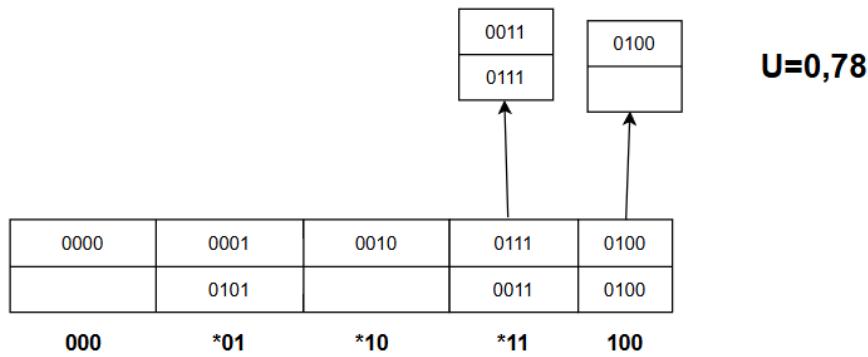
Με την 3η προσθήκη:



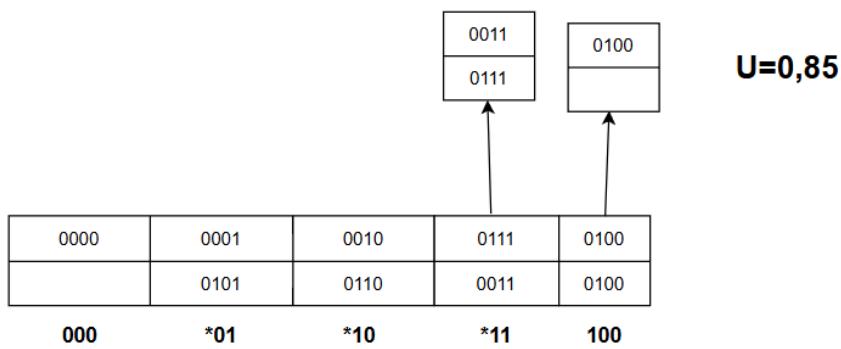
Μετά την 4η προσθήκη:



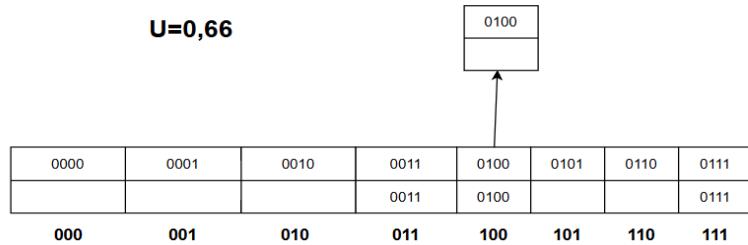
Μετά την 5η προσθήκη:



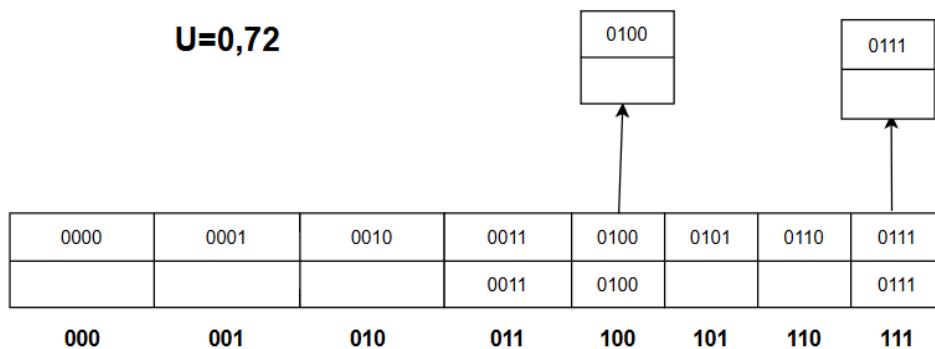
Μετά την 6η προσθήκη το Ο είναι μεγαλύτερο του 80% όποτε κάνουμε πάλι σπλιτ:



Μετά το 2o split:



Μετά την τελευταία προσθήκη:



4γ) Άμα το κλειδί που ψάχνουμε βρίσκεται στην στα πρώτα bucket θα βρεθεί με την πρώτη προσπέλαση ενώ αν βρισκεται σε overflow bucket βρίσκεται με την δευτερη. Έχουμε 13 εγγραφές σε δέκα κάδους οπότε πιθανότητα να βρεθεί στα πρώτα Bucket $10/13=0,76$ πιθανότητα να βρεθεί στον πρώτο κάδο. Η πιθανότητα να βρεθεί στον κάδο υπερχείλισης είναι $1-0,76=0,24$. Οπότε προκύπτει ότι ο μέσος όρος προσπελάσεων για την ανάκτηση μια εγγραφής είναι $1 * 0,76 + 2 * 0,24 = 1,24$ προσπελάσεις.

Άσκηση 5:

Στο δέντρο kd-tree, κάθε κόμβος διαχωρίζει τα δεδομένα με βάση μία διάσταση. Στην συγκεκριμένη περίπτωση οι διαστάσεις είναι salary και age.

α) $\text{Salary} > 150 \wedge \text{Age} \leq 47 \wedge \text{Salary} \leq 300$

Για $\text{salary}=260$ Και $\text{Age}=30$

Αρχικά γίνεται η σύγκριση για $\text{salary}=150$. Έχουμε $\text{Salary}=260$ οπότε πάμε δεξιά. Μετά γίνεται η σύγκριση για $\text{age}=47$. Έχουμε $\text{age}=30$ οπότε πάμε πάλι αριστερά. Τέλος γίνεται σύγκριση για $\text{Salary}=300$ οπότε πάμε αριστερά που βρίσκουμε και το αποτέλεσμα που ψάχνουμε.

β,γ) $\text{Salary} \leq 150 \wedge \text{Age} \leq 60 \wedge \text{Salary} > 80$

Για $\text{salary}=100$ Και $\text{Age}=50$ και Για $\text{salary}=120$ Και $\text{Age}=50$

Αρχικά γίνεται η σύγκριση για $\text{salary}=150$. Έχουμε $\text{salary}=100$ και 120 οπότε πάμε αριστερά. Μετά γίνεται η σύγκριση για $\text{age}=60$. Και στις δύο περιπτώσεις έχουμε $\text{age}=50$ οπότε πάμε πάλι αριστερά. Τέλος γίνεται σύγκριση για $\text{Salary}=80$ οπότε πάμε δεξιά που βρίσκουμε και τα επιθυμητά αποτελέσματα αντίστοιχα.