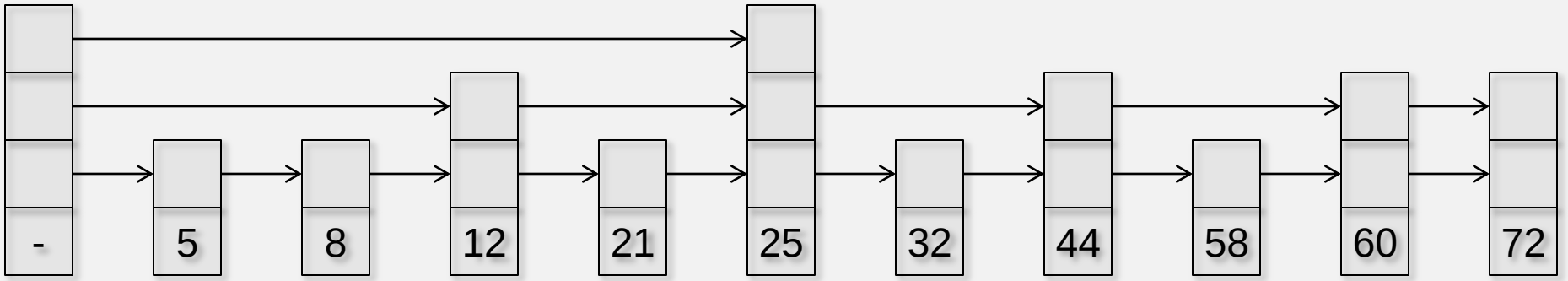
Λίστες παράλειψης (skip lists)

Χρησιμοποιεί πρόσθετους συνδέσμους στους κόμβους μιας συνδεδεμένης λίστας

$\Rightarrow$ επιτάχυνση της αναζήτησης με παράλειψη μεγάλων τμημάτων της λίστας

Μια λίστα παράλειψης είναι μια διατεταγμένη συνδεδεμένη λίστα της οποίας κάθε κόμβος περιέχει μεταβλητό πλήθος συνδέσμων.

Οι σύνδεσμοι του επιπέδου i υλοποιούν απλά συνδεδεμένες λίστες οι οποίες παραλείπουν τους κόμβους με λιγότερους από i συνδέσμους.



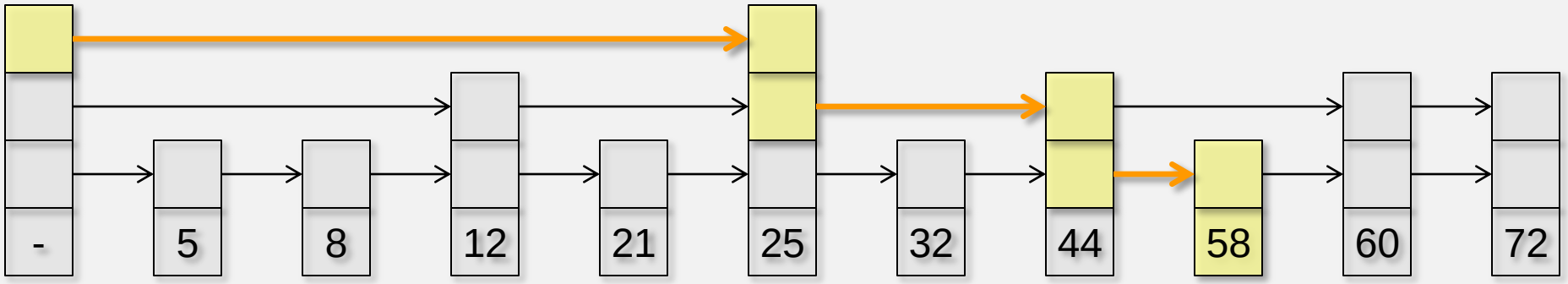
Λίστες παράλειψης (skip lists)

Χρησιμοποιεί πρόσθετους συνδέσμους στους κόμβους μιας συνδεδεμένης λίστας

$\Rightarrow$ επιτάχυνση της αναζήτησης με παράλειψη μεγάλων τμημάτων της λίστας

Μια λίστα παράλειψης είναι μια διατεταγμένη συνδεδεμένη λίστα της οποίας κάθε κόμβος περιέχει μεταβλητό πλήθος συνδέσμων.

Οι σύνδεσμοι του επιπέδου i υλοποιούν απλά συνδεδεμένες λίστες οι οποίες παραλείπουν τους κόμβους με λιγότερους από i συνδέσμους.

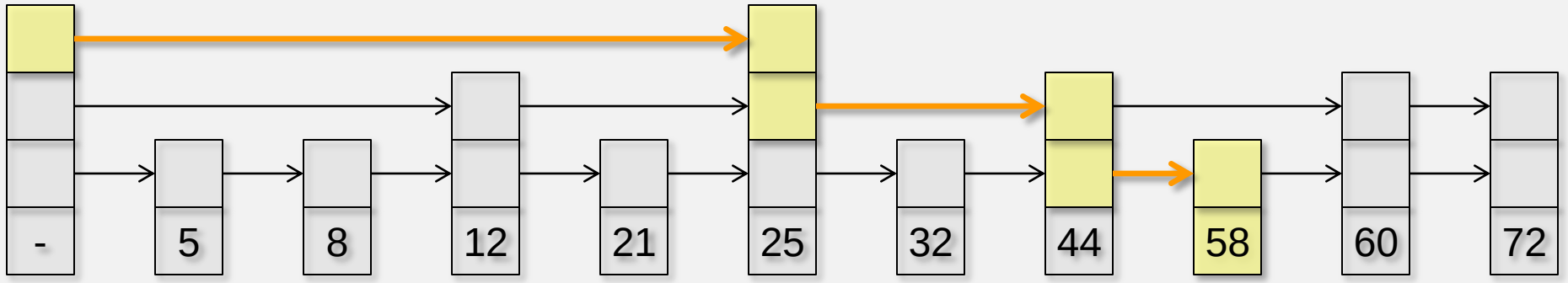


Αναζήτηση 58

Λίστες παράλειψης (skip lists)

επίπεδο αναζήτησης

```
private Item searchR(Node t, Key v, int k)
{
   if (t==null) return null;
   if (t!=head)
      if (v.equals(t.key)) return t.item;
   if (k>=t.size) k = t.size-1;
   if (t.next[k] != null) {
      if ( v.compareTo(t.next[k].key) > 0 )
         return searchR(t.next[k], v, k);     // επόμενος κόμβος του επιπέδου k
   return (k==0) ? null : searchR(t, v, k-1); // αναζήτηση στο επίπεδο k-1
}

Item search(Key v) { return searchR(head, v, lgN-1); }
```



Αναζήτηση 58

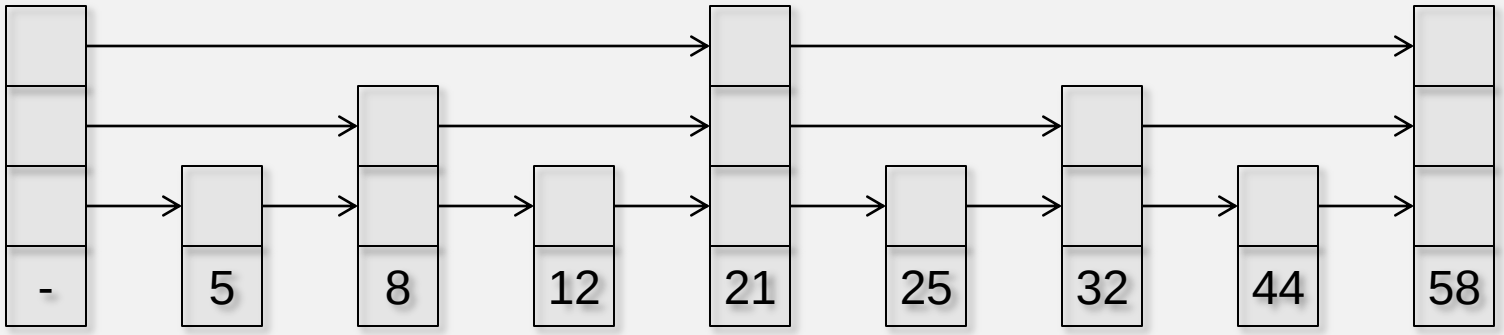
Λίστες παράλειψης (skip lists)

Λίστα παράλειψης με παράμετρο t

- Κάθε t κόμβους υπάρχει ένας με ≥ 2 συνδέσμους
- Κάθε t^2 κόμβους υπάρχει ένας με ≥ 3 συνδέσμους

$\vdots$

- Κάθε t^j κόμβους υπάρχει ένας με $\geq j+1$ συνδέσμους



Λίστες παράλειψης (skip lists)

Λίστα παράλειψης με παράμετρο t

- Κάθε t κόμβους υπάρχει ένας με ≥ 2 συνδέσμους
- Κάθε t^2 κόμβους υπάρχει ένας με ≥ 3 συνδέσμους

$\vdots$

- Κάθε t^j κόμβους υπάρχει ένας με $\geq j+1$ συνδέσμους

Αν έχουμε N κόμβους τότε $t^j \leq N \Rightarrow j \leq \log_t N = \dfrac{\lg N}{\lg t}$

Λίστες παράλειψης (skip lists)

Λίστα παράλειψης με παράμετρο t

- Κάθε t κόμβους υπάρχει ένας με ≥ 2 συνδέσμους
- Κάθε t^2 κόμβους υπάρχει ένας με ≥ 3 συνδέσμους

$\vdots$

- Κάθε t^j κόμβους υπάρχει ένας με $\geq j+1$ συνδέσμους

Αν έχουμε N κόμβους τότε $t^j \leq N \Rightarrow j \leq \log_t N = \frac{\lg N}{\lg t}$

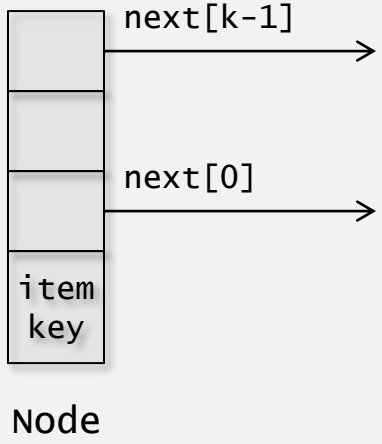
Τυχαιοποιημένη λίστα παράλειψης

Ένας κόμβος έχει $j+1$ συνδέσμους με πιθανότητα $1/t^j$

Λίστες παράλειψης (skip lists)

Αρχικοποίηση λίστας παράλειψης

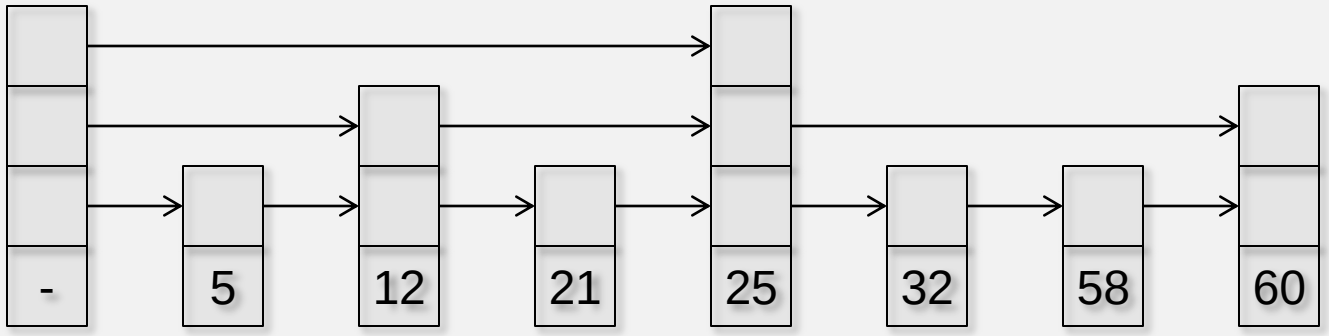
```
public class SkipList<Key extends Comparable<Key>, Item> {

   private class Node {
      Item item; Key key;
      Node[] next;         // πίνακας συνδέσμων
      int size;            // επίπεδο κόμβου
      Node(Item u, Key v, int k)
      { item = u; key = v; size = k; next = new Node[size]; }

   private Node head;
   private int N, lgN;
   private static final int L = 50; // μέγιστο πλήθος επιπέδων

   SkipList(int maxN)
   {
      N = 0; lgN = 0;
      head = new Node(null, null, L);
   }
}
```



Λίστες παράλειψης (skip lists)

Εισαγωγή σε λίστα παράλειψης

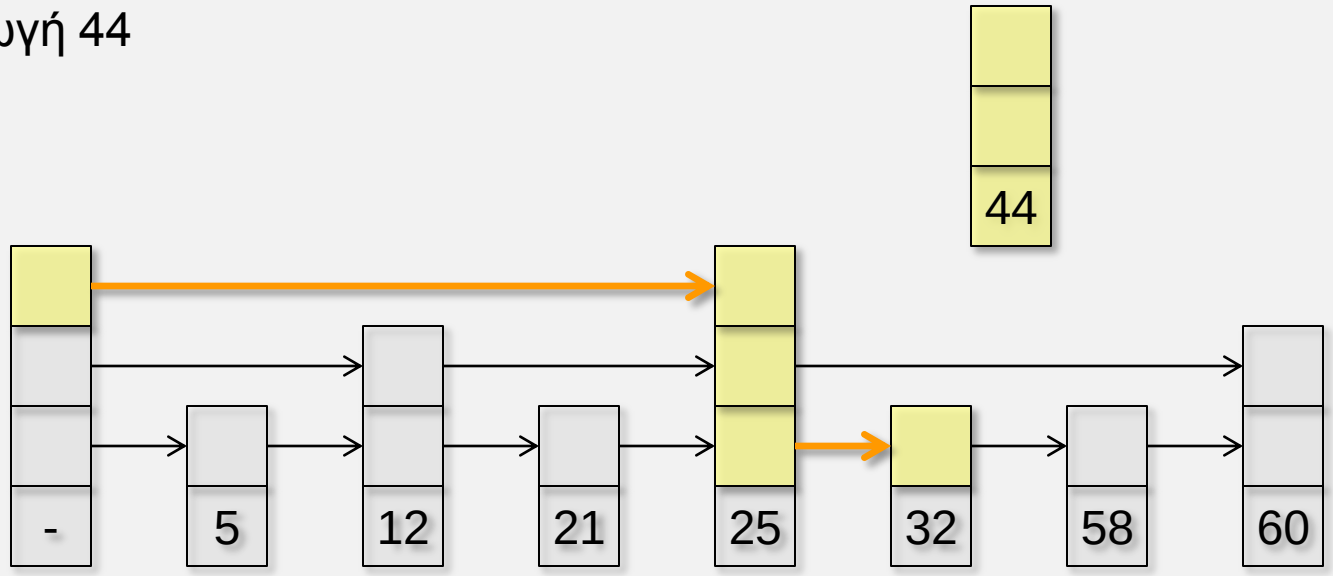
Εισαγωγή 44



Λίστες παράλειψης (skip lists)

Εισαγωγή σε λίστα παράλειψης

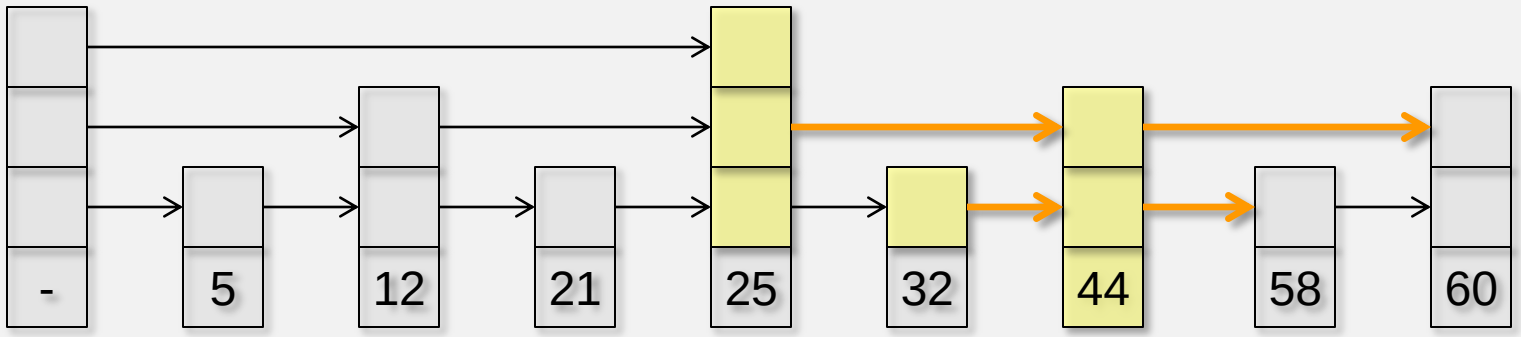
Εισαγωγή 44



Λίστες παράλειψης (skip lists)

Εισαγωγή σε λίστα παράλειψης

Εισαγωγή 44



Λίστες παράλειψης (skip lists)

Εισαγωγή σε λίστα παράλειψης με παράμετρο $t = 2$

```
private int randX() {
   int i, j; double t = Math.random();
   for (i = 1; j = 2; i < L; i++, j += j)
      if (t*j > 1.0) break;
   if (i > lgN) lgN = i;
   return i;
}
```

τυχαία επιλογή επιπέδου του νέου κόμβου

```
private void insertR(Node t, Node x, int k) {
   Key v = x.key; Node tk = t.next[k];
   if ( (tk == null) || (v.compareTo(tk.key) < 0) ) {
      if (k < x.size) { x.next[k] = tk; t.next[k]=x; }
      if (k == 0) return;
      insertR(t, x, k-1); return; // ένα επίπεδο πιο κάτω
   }
   insertR(tk, x, k); // επόμενος κόμβος, ίδιο επίπεδο
}

public void insert(Item u, Key v)
{ insertR(head, new Node(u, v, randX()), lgN); N++; }
```

Λίστες παράλειψης (skip lists)

Λίστα παράλειψης με παράμετρο t

Ιδιότητα: Η αναζήτηση και η εισαγωγή σε μια τυχαιοποιημένη λίστα παράλειψης με παράμετρο t απαιτούν κατά μέσο όρο περίπου $\frac{t \log_t N}{2} = \frac{t}{2 \lg t} \lg N$ συγκρίσεις

Ιδιότητα: Μια τυχαιοποιημένη λίστα παράλειψης με παράμετρο t έχει κατά μέσο όρο $\frac{t}{t-1} N$ συνδέσμους

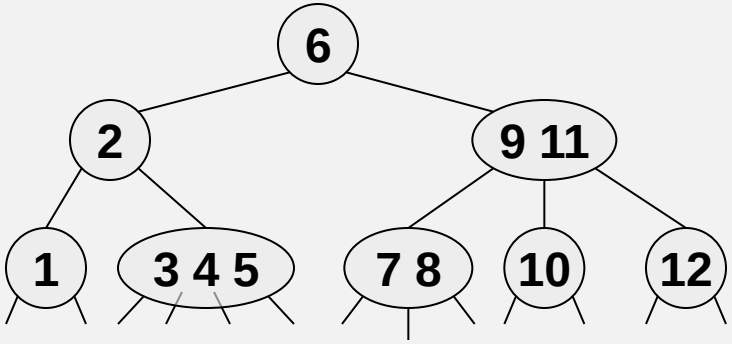
Λίστες παράλειψης (skip lists)

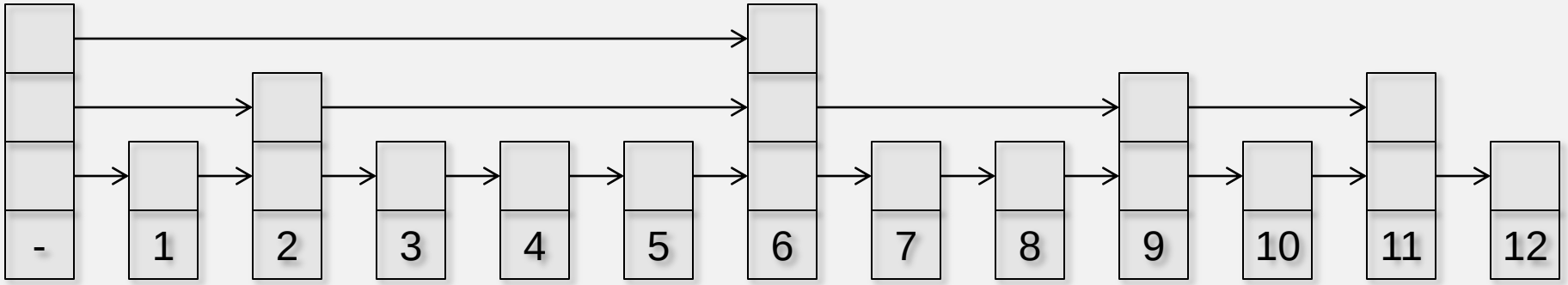
Διαγραφή από λίστα παράλειψης

```
private void deleteR(Node t, Key v, int k)
{
   Node x = t.next[k];
   if ( v.compareTo(x.key) <= 0 )
   {
      if ( v.equals(x.key) ) { t.next[k] = x.next[k]; // διαγραφή του κόμβου x }
      if (k == 0) return;
      deleteR(t, v, k-1); return; // ένα επίπεδο πιο κάτω
   }
   deleteR(t.next[k], v, k); // επόμενος κόμβος, ίδιο επίπεδο
}

public void delete(Key v)
{
        deleteR(head, v, lgN);
        N--;
}
```

Λίστες παράλειψης (skip lists)

Αναπαράσταση (2,4)-δένδρου με λίστα παράλειψης





Κάθε κόμβος έχει αριθμό συνδέσεων ίσο με το ύψος του στο (2,4)-δένδρο