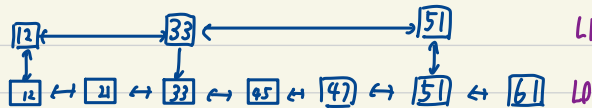<u>skip list pros:</u>

1. easy implement

2. maintain a dynamic set of n elements
   in $O(\lg n)$ time per operation in excepted running time

<u>Review:</u>　Linked List:　$\llcorner\square\leftrightarrow\square\leftrightarrow\square\leftrightarrow\square\to Nil$

　　　　　　　Search: $O(n)$

　　Two Sorted Linked List:

　　　　　　• minimize the distance travel.

　　　　Search for 47:　　12 → 33 → 33 → 45 → 47
　　　　　　　　　　　　　　　L1　　L1　　L0　　L0　　L0

• How to structure double sorted linked list?
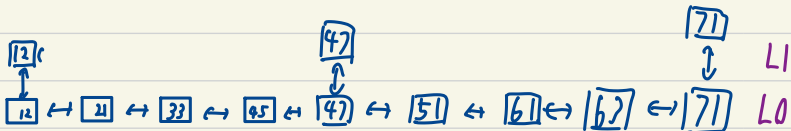

　　Search (x) [2 linked list]
　　　　　在 L1 走訪至不起過 x 之最大 node
　　　　　走訪至 L2
　　　　　最終走訪至 x, 或找不到 x

<u>Analysis:</u>　　Search cost $\simeq |L_1| + \dfrac{|L_0|}{|L_1|} = \theta(2\sqrt{n})$

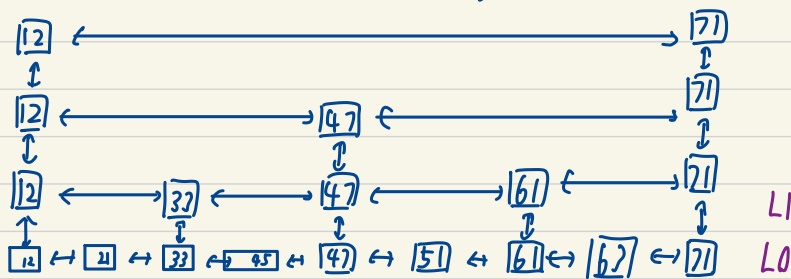　　　minimize this: when $|L_1|^2 = |L_0| = n \Rightarrow |L_1| = \sqrt{n}$

　　　　not enough! $\Rightarrow O(\lg n)$

• two Sorted Linked List $\Rightarrow O(2\sqrt{n})$
• three Sorted Linked List $\Rightarrow O(3\sqrt[3]{n})$

- k   Sorted Linked List ⇒ O(k $\sqrt[k]{n}$)
- lgn Sorted Linked List ⇒ O(lgn $\sqrt[lgn]{lgn}$) = O(2 lgn) ⇒ 只適用於 static set

Insert (x)
先用 Search(x) 找到 x 在 bottom list 的合適位置
並插入 某些 其它的 list (用 flip fair coin 決定)

flip fair coin
可能已經 created

if Heads, promote x to next level up and repeat
else stop

Note:
promote level up 可能會 create new linked list ⇒ t增加 level of linked list
⇒ messy ⇒ randomized search cost

worst case:
一直是 Head ⇒ ∞ ?

Delete (x):
先用 Search(x) 找到 x, 把 x 的所有 level 刪掉

Warmup Lemma:
of levels in n-element skip list is O(lgn) with high probability ⇒ prob: 1 - $\frac{1}{n^\alpha}$, where α > 1