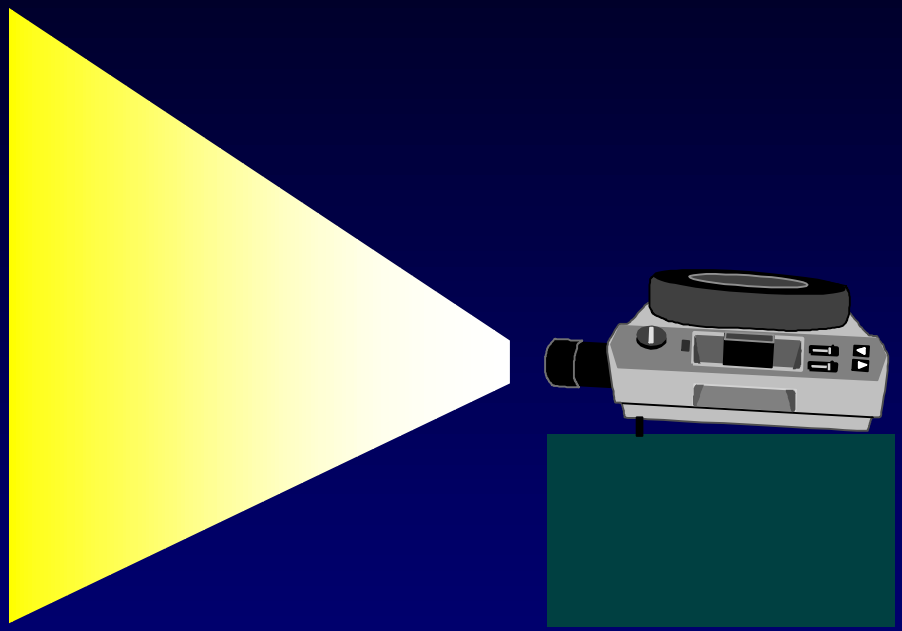
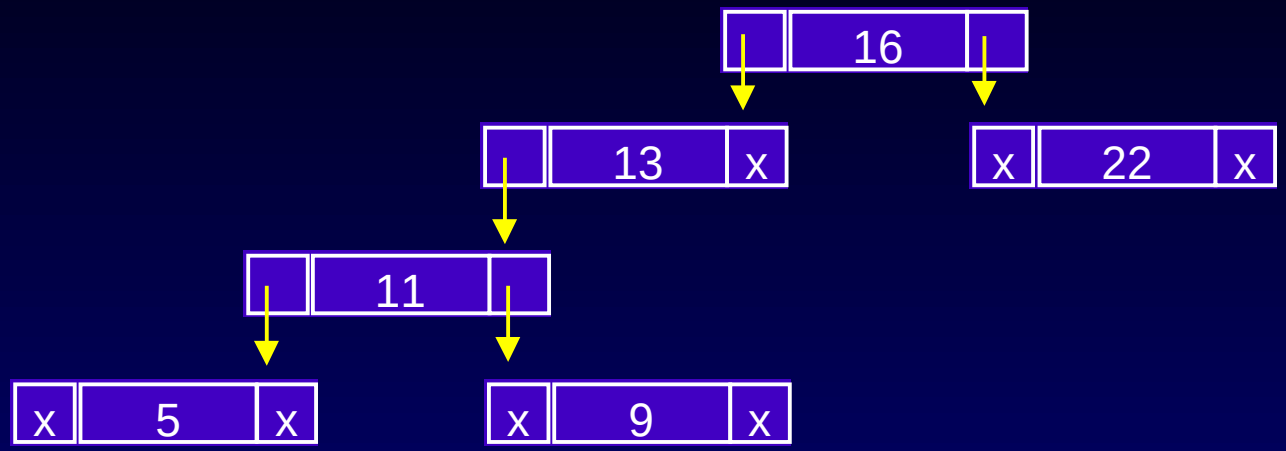
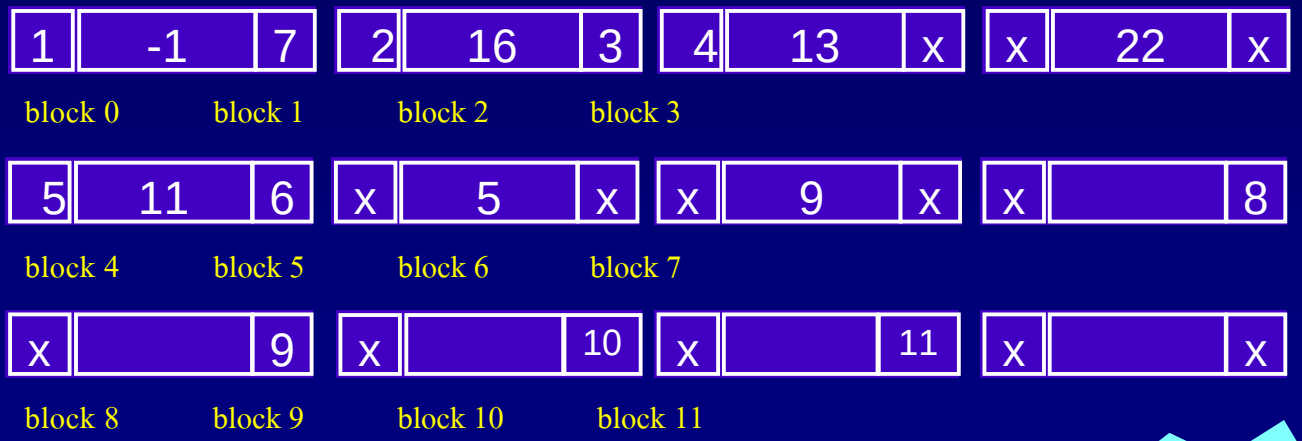
Tree-Based Files : Outline

- Binary Search Trees
- Threaded BST
- Paged Trees
- B-Trees
- B +-Trees

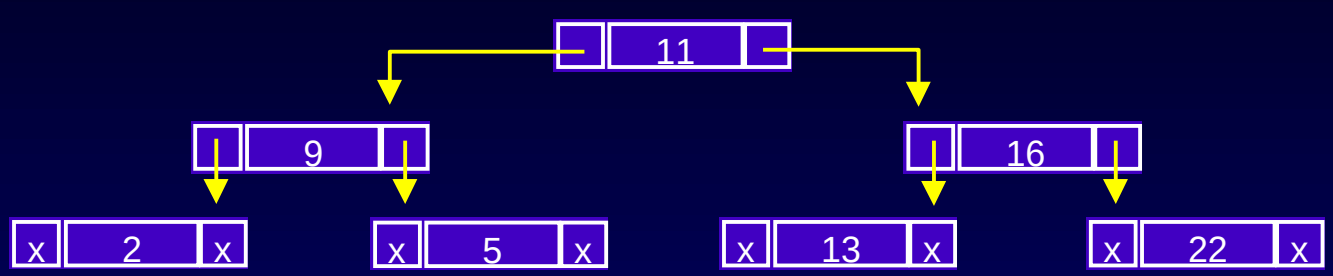


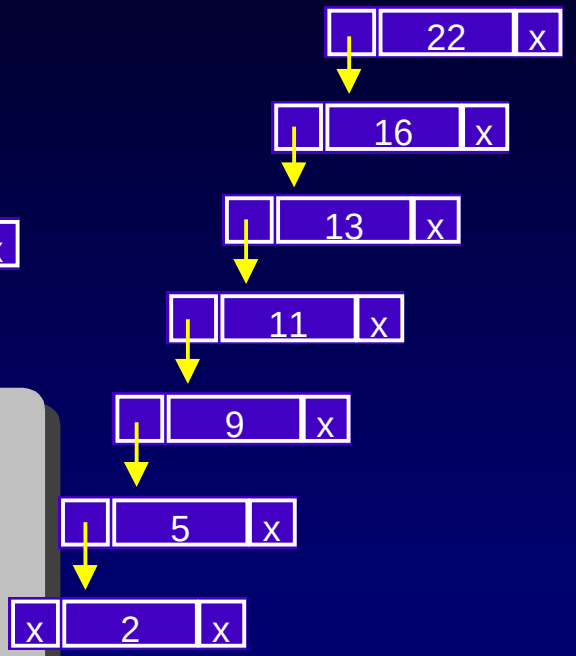
Binary Search Trees





Binary Search Trees





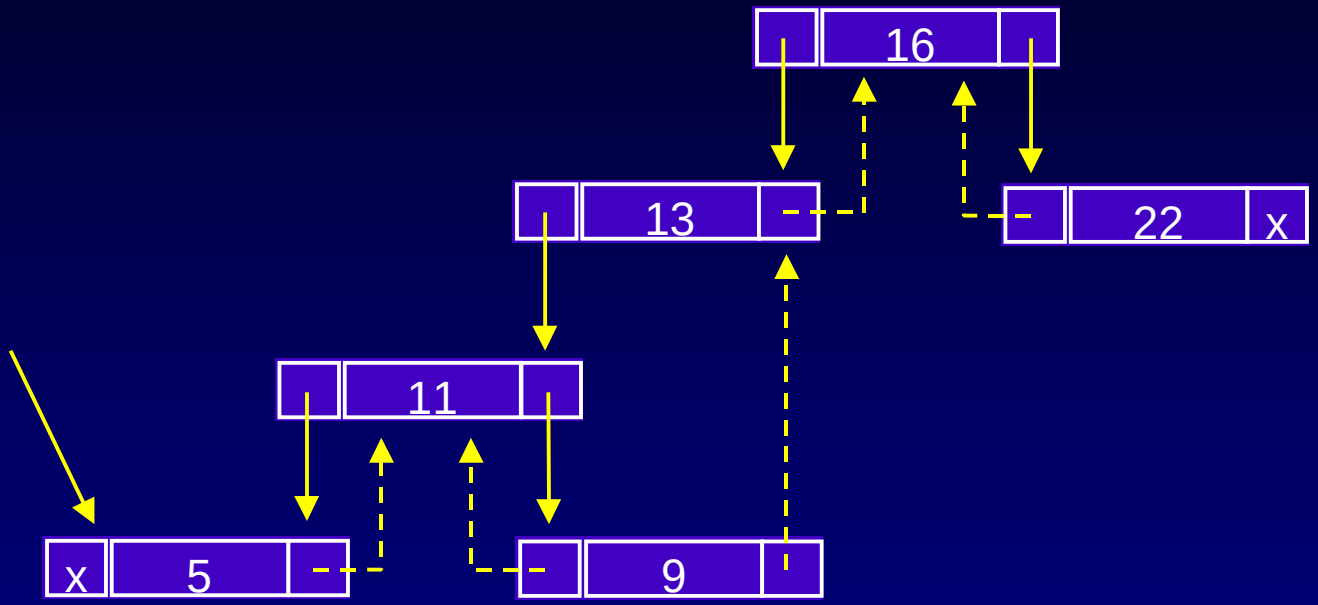
Worst case :

$$SL_{avg}[RetrieveOne, BST] = \frac{\sum_{k=1}^{NR} k}{NR} = \frac{NR \cdot (NR+1)}{2 \cdot NR} = \frac{NR+1}{2} \quad rba$$

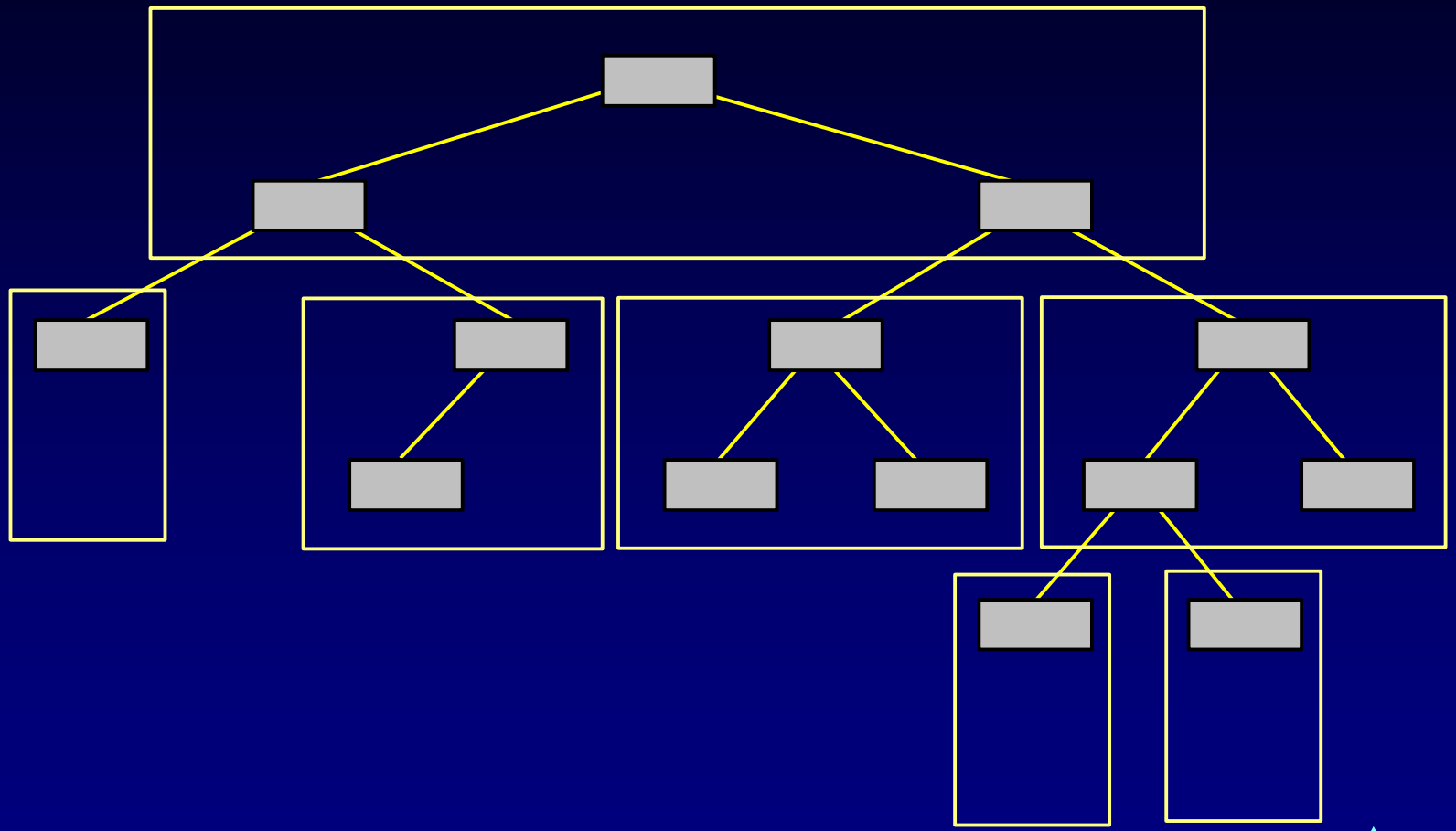
Best case :

$$SL_{avg}[RetrieveOne, BST] = \frac{\sum_{k=1}^{m} \left[k \cdot 2^{(k-1)} \right]}{NR} = \log_2 \left[NR+1 \right]_{-1} \quad rba$$

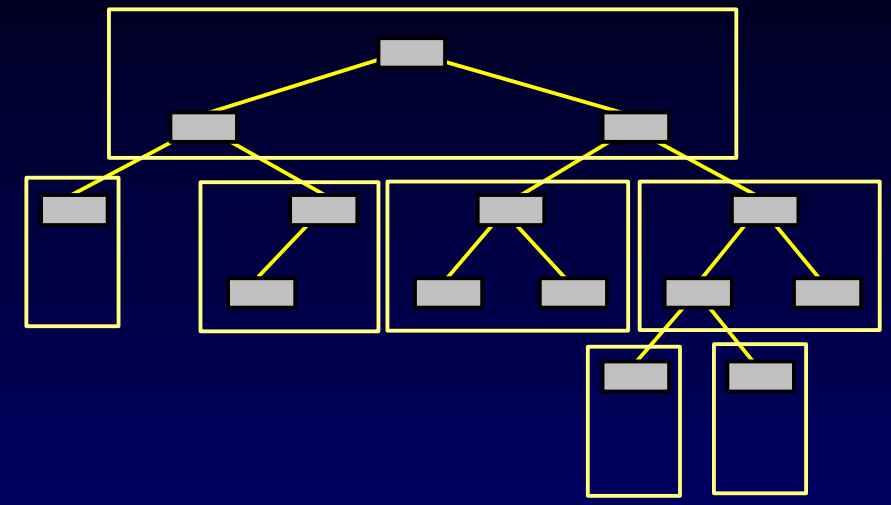
Threaded Binary Search Trees



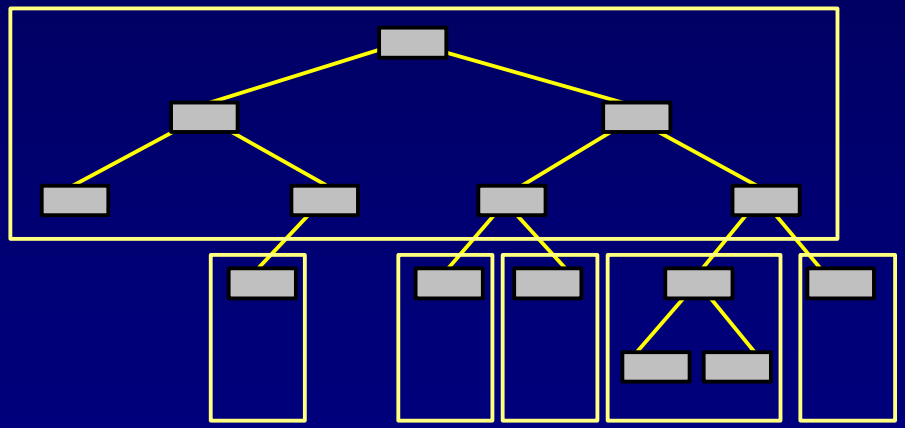
Paged Trees



Paged Trees

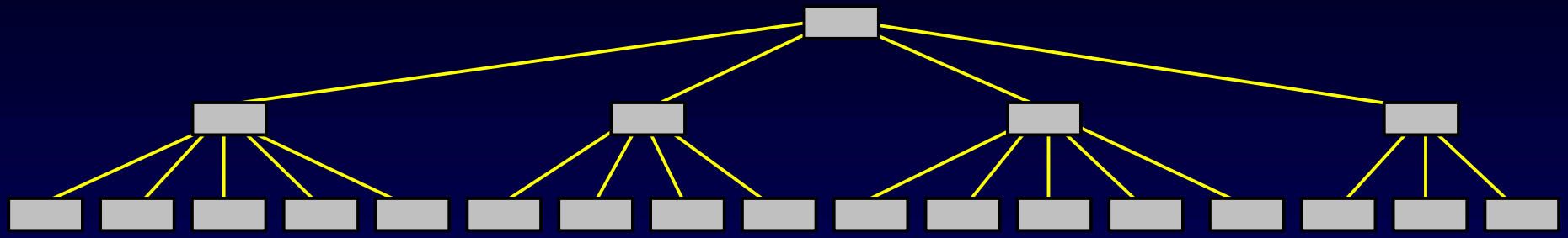


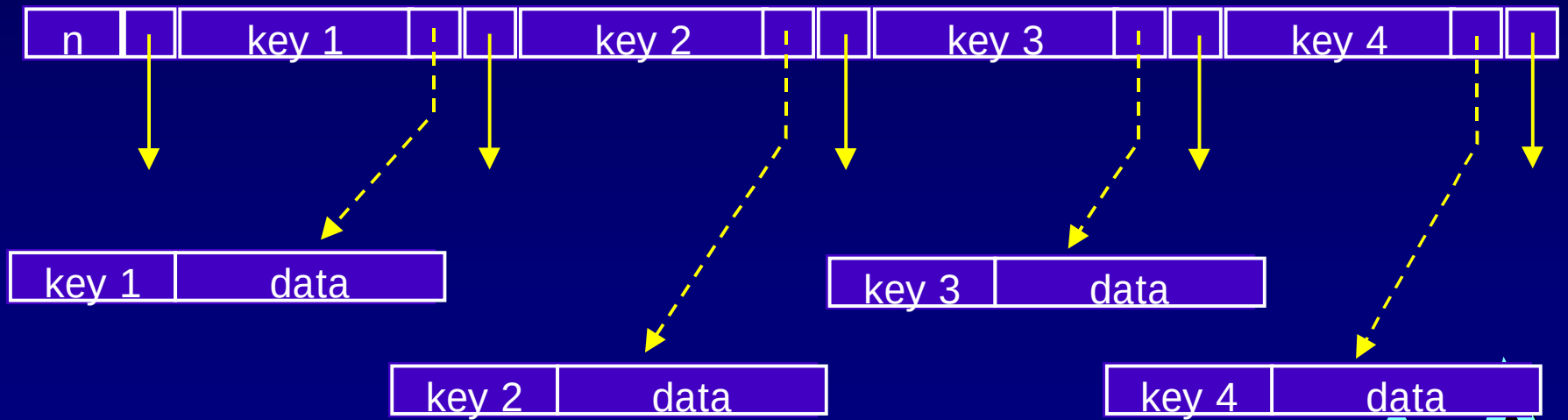
66%



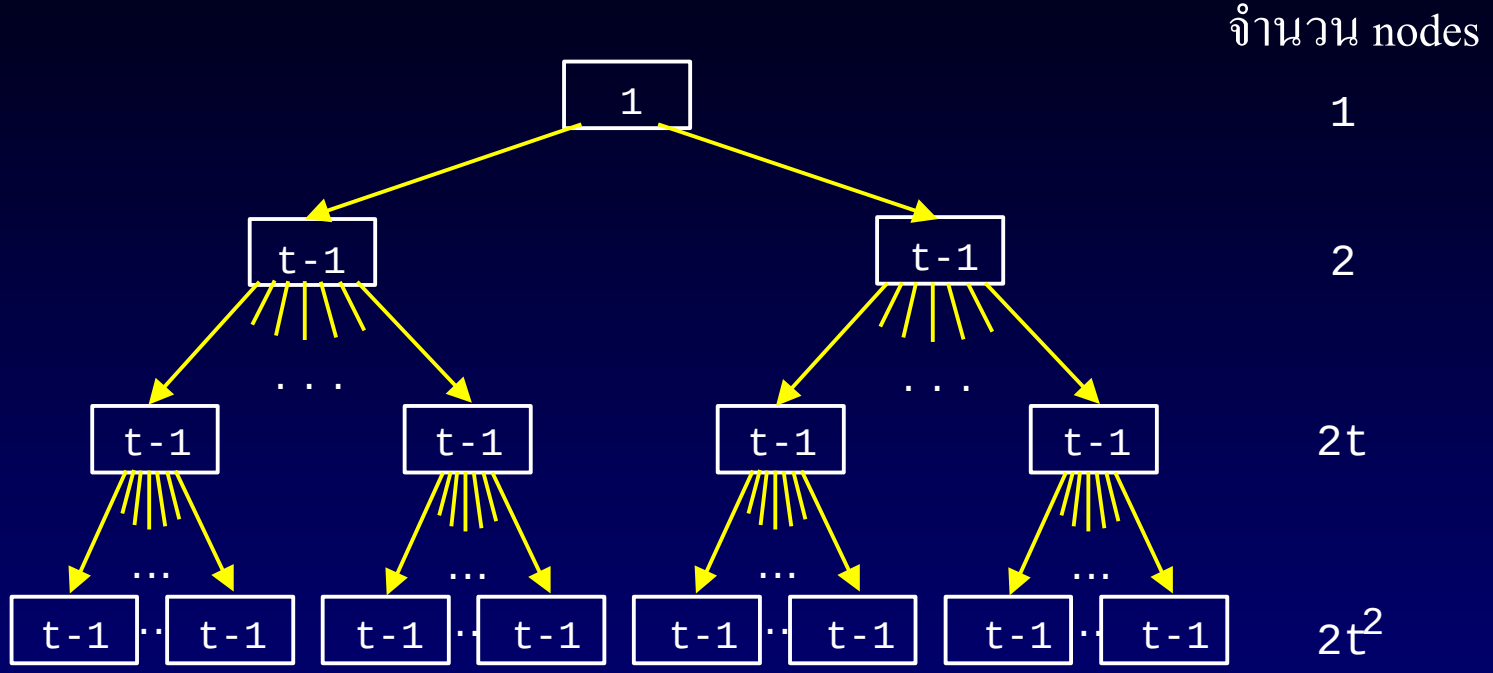
33%

B-Trees





The Height of a B-Tree



$$n \geq 1 + (t-1)\sum_{i=1}^{h} 2t^{i-1} = 1 + 2(t-1)\left(\frac{t^h - 1}{t-1}\right) = 2t^h - 1$$

$$h \leq \log_t \left(\frac{n+1}{2}\right) \text{ if } t = \left\lceil \frac{m}{2} \right\rceil$$

Searching a B-Tree : Analysis

สมมติว่าเวลาในการอ่าน 1 node ของ B-tree ที่มี order m จาก disk คือ

$$n \geq 1+(t-1)\sum_{i=1}^{h} 2t^{i-1} = 1+2(t-1)\left(\frac{t^h-1}{t-1}\right) = 2t^h-1$$

$$h \leq \log_t\left(\frac{n+1}{2}\right) \text{ if } t=\left\lfloor \frac{m}{2} \right\rfloor$$

โดยใช้ binary search ในการหาคีย์ที่ต้องการใน node ที่อ่านได้ ซึ่งใช้เวลา

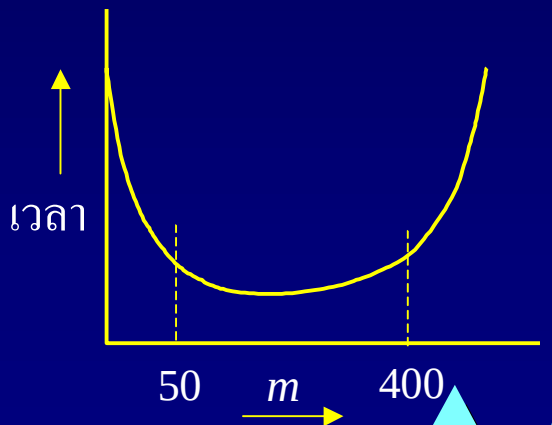
$$n \geq 1+(t-1)\sum_{i=1}^{h} 2t^{i-1} = 1+2(t-1)\left(\frac{t^h-1}{t-1}\right) = 2t^h-1$$

$$h \leq \log_t\left(\frac{n+1}{2}\right) \text{ if } t=\left\lfloor \frac{m}{2} \right\rfloor$$

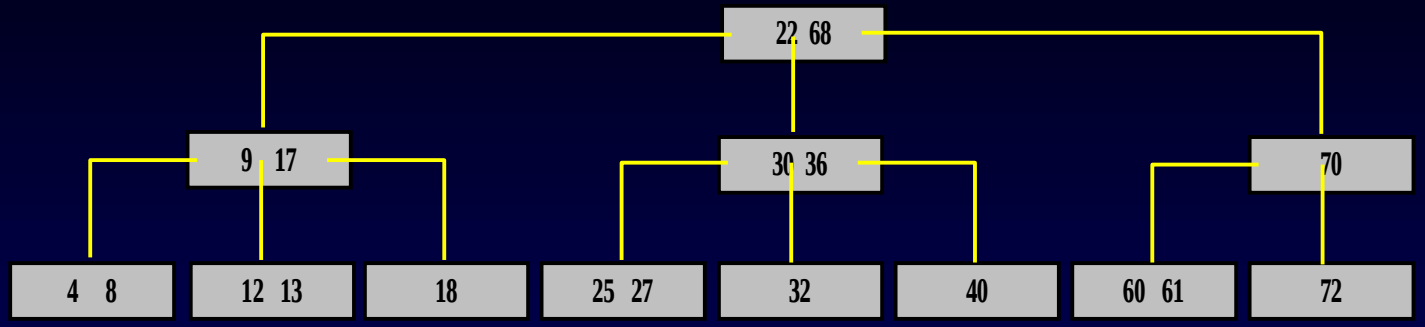
และจำนวน nodes ที่ต้องอ่าน (worst case) ใน B-tree (ซึ่งมี n nodes) เพื่อหาคีย์ที่ต้องการ จะเท่ากับความสูงของต้นไม้ h ดังนั้นเวลาในการหาคีย์คือ

$$n \geq 1+(t-1)\sum_{i=1}^{h} 2t^{i-1} = 1+2(t-1)\left(\frac{t^h-1}{t-1}\right) = 2t^h-1$$

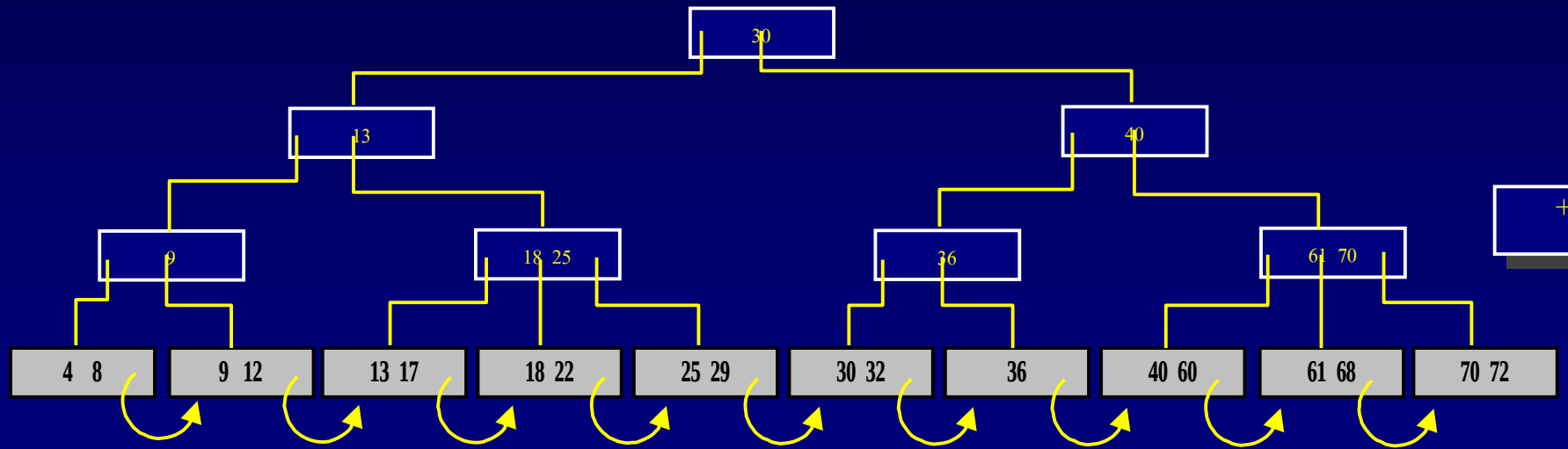
$$h \leq \log_t\left(\frac{n+1}{2}\right) \text{ if } t=\left\lfloor \frac{m}{2} \right\rfloor$$



B$^+$-Trees

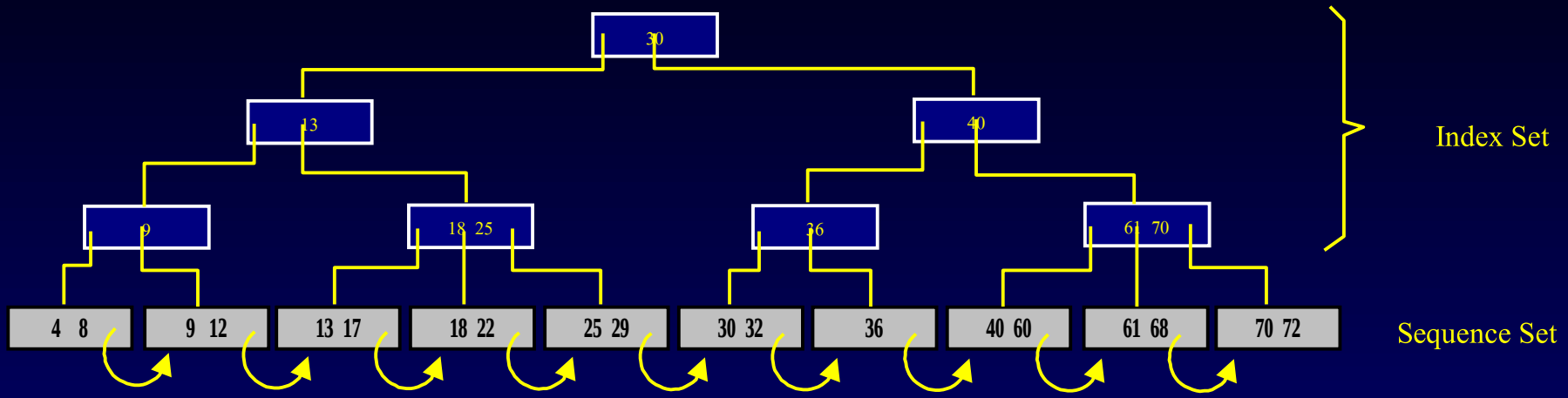


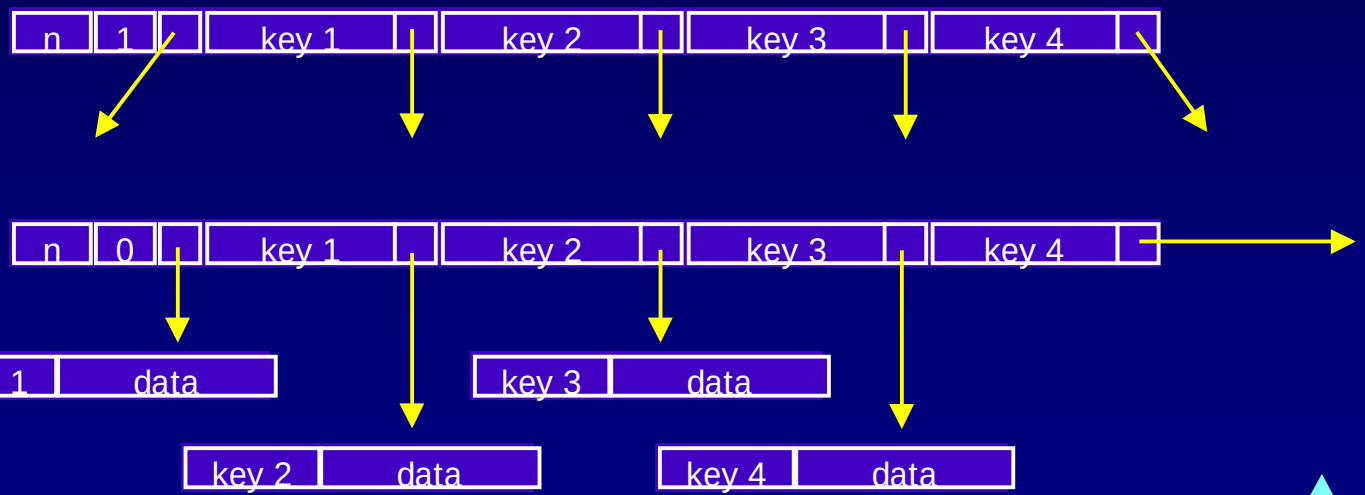
B-tree



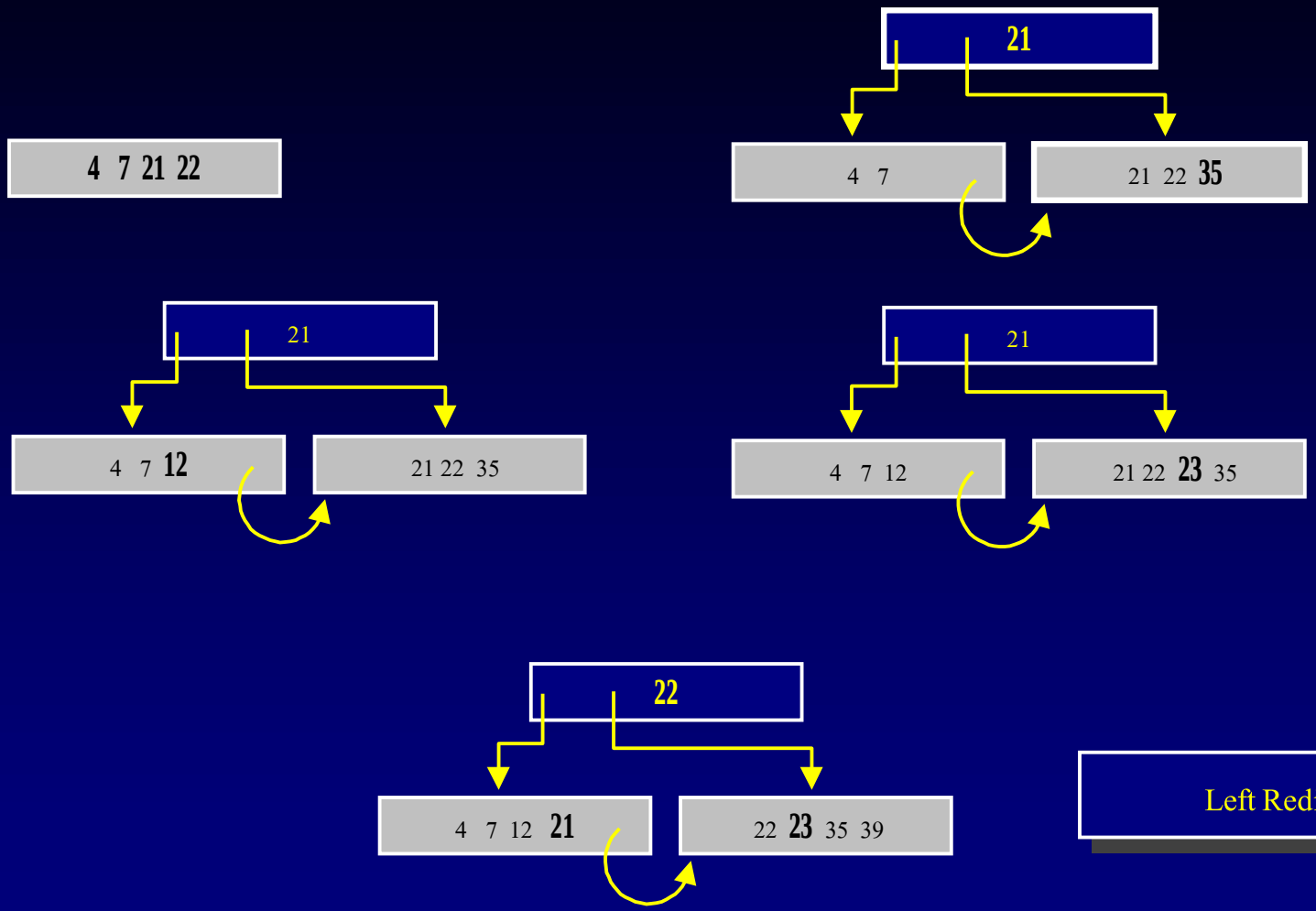
B$^+$-tree

B+-Trees

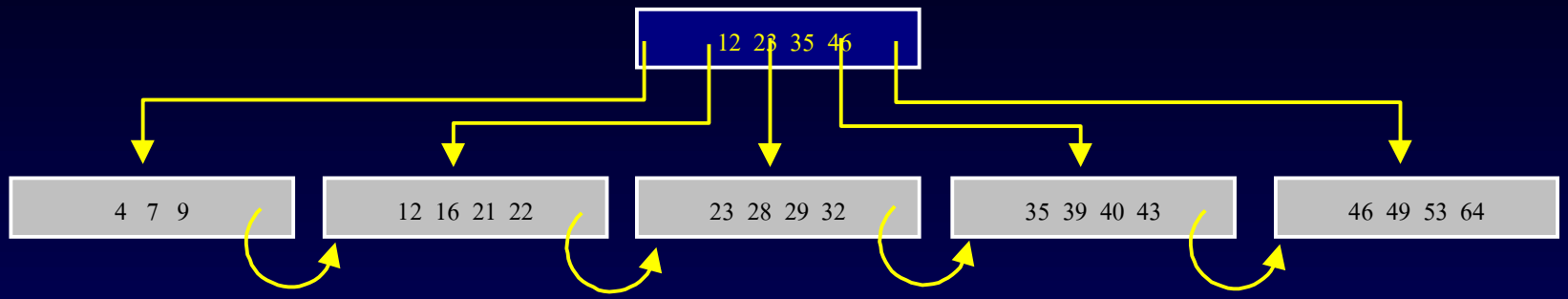

30
13
40
9
18 25
36
61 70
4 8
9 12
13 17
18 22
25 29
30 32
36
40 60
61 68
70 72
Index Set
Sequence Set


n
1
key 1
key 2
key 3
key 4
n
0
key 1
key 2
key 3
key 4
key 1
data
key 3
data
key 2
data
key 4
data

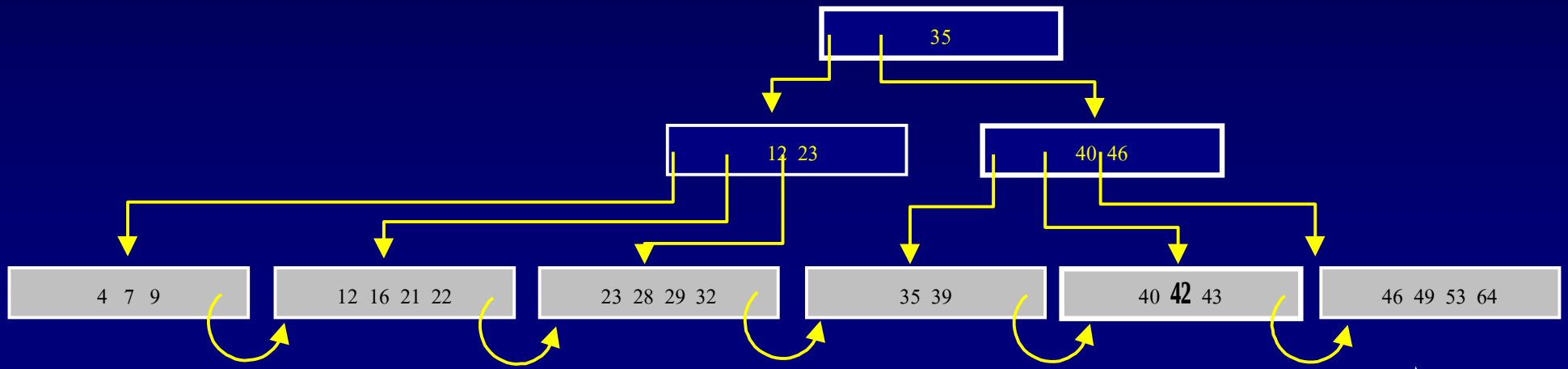
B$^+$-Trees : Insert



B$^+$-Trees : Insert



Insert : 16, 32, 29, 46, 28, 43, 64, 9, 49, 53, 40



B$^+$-Trees : Initial Loading