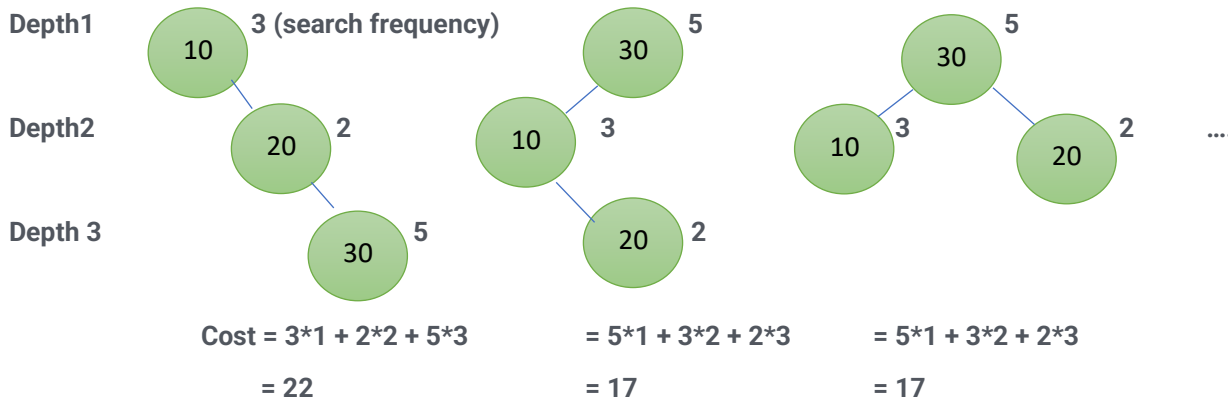


Optimal BST

Every key has a different probability to be searched for.



Brute force solution: build all tree configurations that contain the keys and see which is best.

Efficient solution: Dynamic Programming

B-Tree

How can we store a tree into a file?

There are two definitions of a B-tree:

Knuth Order (Order) is used by Knuth's definition. The Knuth order m is the maximum number of children. A Knuth order of m means every node must have a $\text{max} = m$, and a $\text{min} = \text{ceil}(m/2)$ children.

CLRS Degree (Degree) is used in the definition in Cormen et al in Introduction to Algorithms (CLRS). The CLRS degree t is the minimum number of children. A CLRS degree of t means every node must have a $\text{min} = t$ and a $\text{max} = 2t$ children. Link to the book: https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf

The number of keys in both cases is equal to the number of children minus one.

!!! Attention: You will find a lot of examples with 'order' of B-Tree, order! = degree

Example:

B-tree of order 5 OR $m=5$

$\text{max children} = m = 5$

$\text{min children} = \text{ceil}(m/2) = 3$

$\text{max keys} = m-1 = 4$

$\text{min keys} = \text{ceil}(m/2)-1 = 2$

B-tree of degree 5 OR $t=5$

$\text{max children} = 2t = 10$

$\text{min children} = t = 5$

$\text{max keys} = 2t-1 = 9$

$\text{min keys} = t-1 = 4$

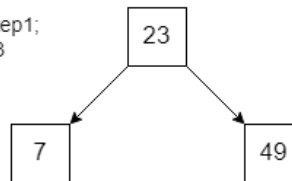
Ex1: Efficient B-Tree insert

t=2; Insert: 49, 7, 23, 4, 16, 9, 51, 28

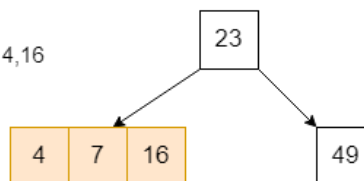
Step1: insert 49, 7, 23



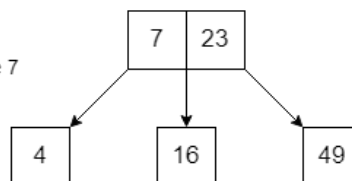
Step2: full node from step1;
split and promote 23



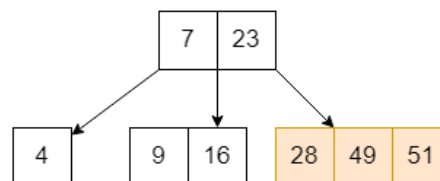
Step3: insert 4, 16



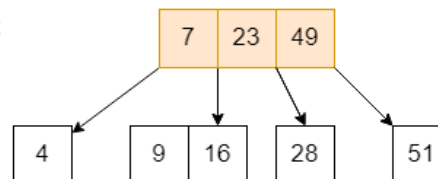
Step4: full node from
step3; split and promote 7



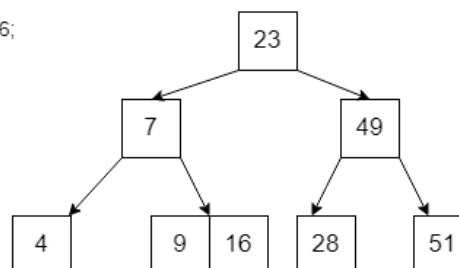
Step5: insert 9, 51



Step6: full node from step5;
split and promote 49



Step7: full node from step6;
split and promote 23

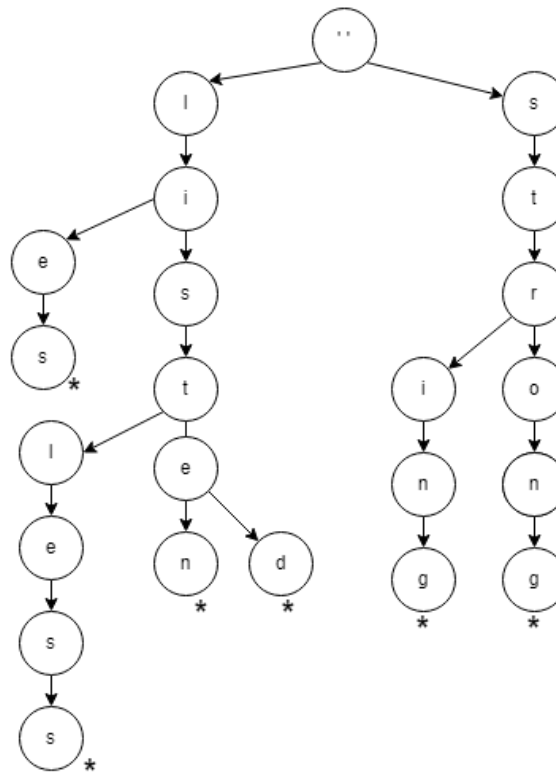


Trie tree

Trie tree insert:

-listen
-listless
-lies
-lists
-listed
-strong
-string

Find listening?



Homework:

1. Build a B-Tree with min degree $t=2$ (each node can contain 1-3 keys) by inserting in order following keys: 35, 2, 1, 89, 4, 33, 24, 6, 7, 88, 14, 23, 22
2. Build a B-Tree with min degree $t=3$ (each node can contain 2-5 keys) by inserting in order following keys: 35, 2, 1, 89, 4, 33, 24, 6, 7, 88, 14, 23, 22, 19
3. Implement a trie tree data structure and add operations to:
 - add a new word to the trie tree
 - print (in alphabetical order) all the words contained in the trie tree

TEST!!!