Math 322

Trees end of 10.1

① All trees $n = i + l$

② full m-ary $n = mi + 1$

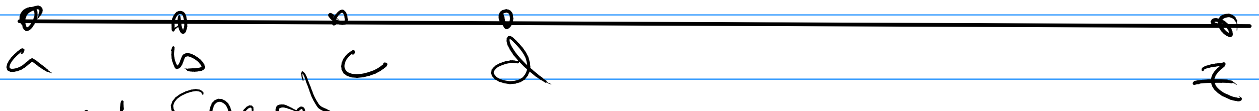
③ All trees $l \le m^h$

④ All trees $\lceil \log_m l \rceil \le h$

⑤ (full m-ary and balanced) $\lceil \log_m l \rceil = h$

10.2 Apps of Trees

① Binary Search Tree (2-ary)



dictionary search

cost = worst case analysis.

happy, hello, Marc, Zoo

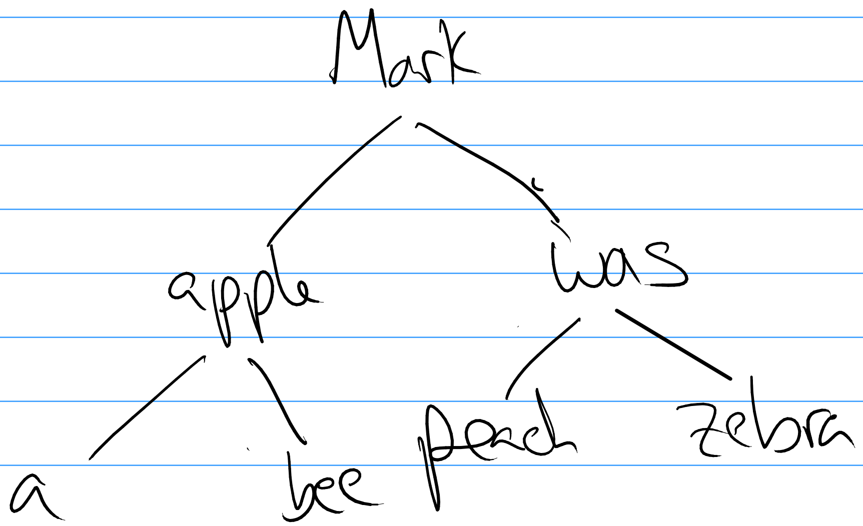
→ every time you have to compare to everyone.

Binary Search Tree

label all vertices.

Mark, was, apple, peach, zebra, a, bee

fast



Cost of Comparison

h + 1

w: "New word"

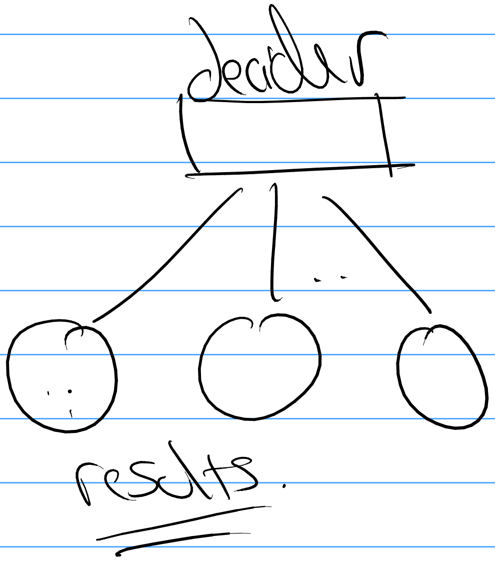
Decision Tree

parent is a decision

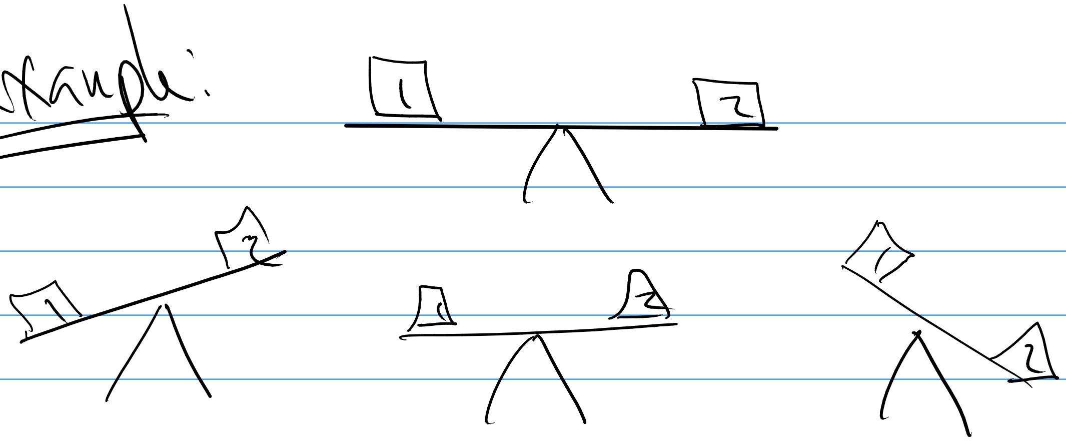
children are outcomes of the decision

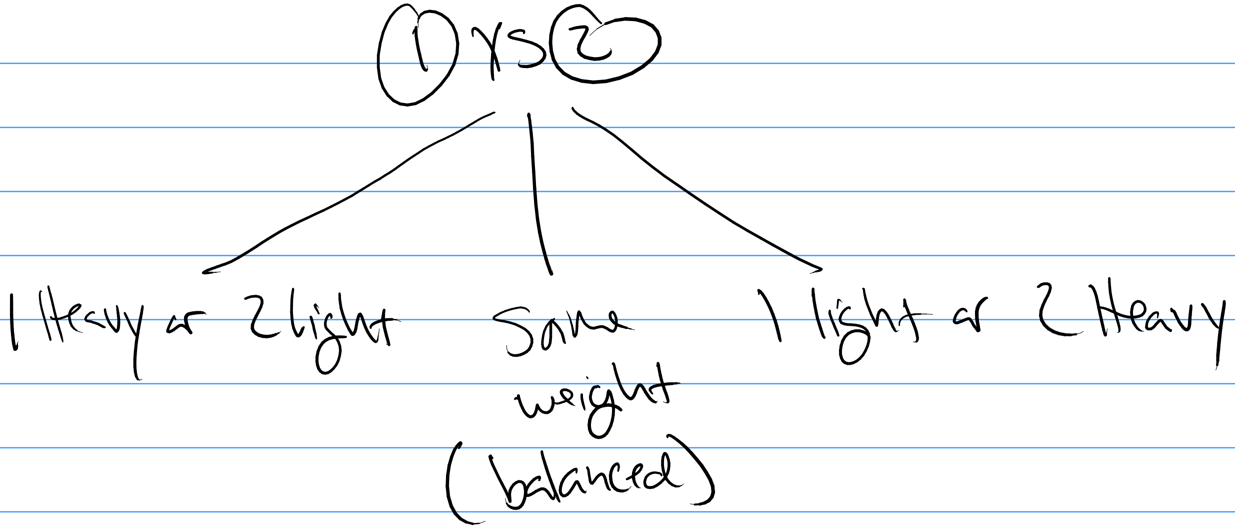
New Decision?

End result?



example:



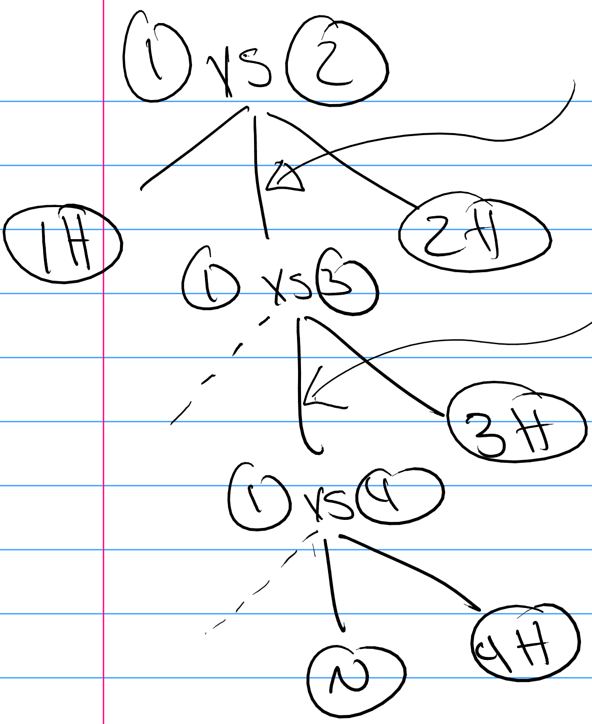


ex 4 coins, 1 maybe a fake, and if you have a fake it is heavy.

Find it?

Possible leaves: 1H, 2H, 3H, 4H or N

no fake



1 and 2 are not fake
3 and 4 may be

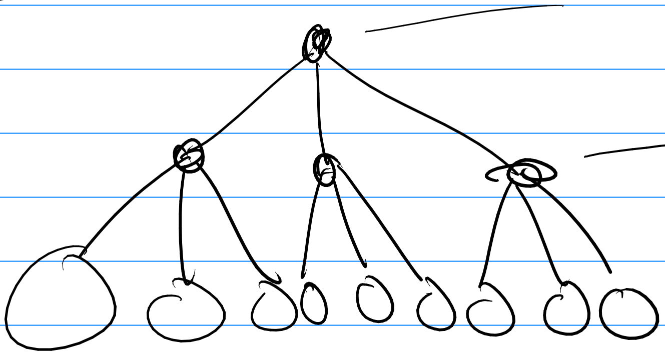
1 and 3 are not fake
4 may be

Can we do better? $l = 5$

$\lceil \log_n l \rceil = h$ full & balanced

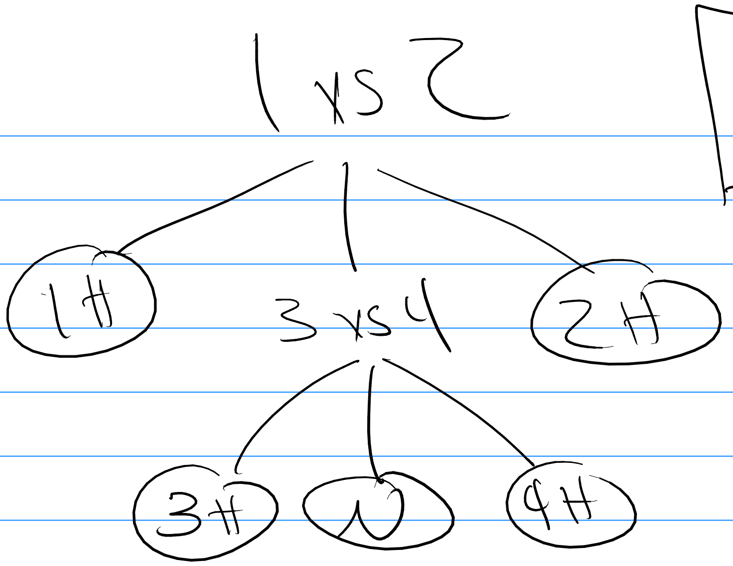
$\lceil \log_3 5 \rceil = 2$

Ideal:



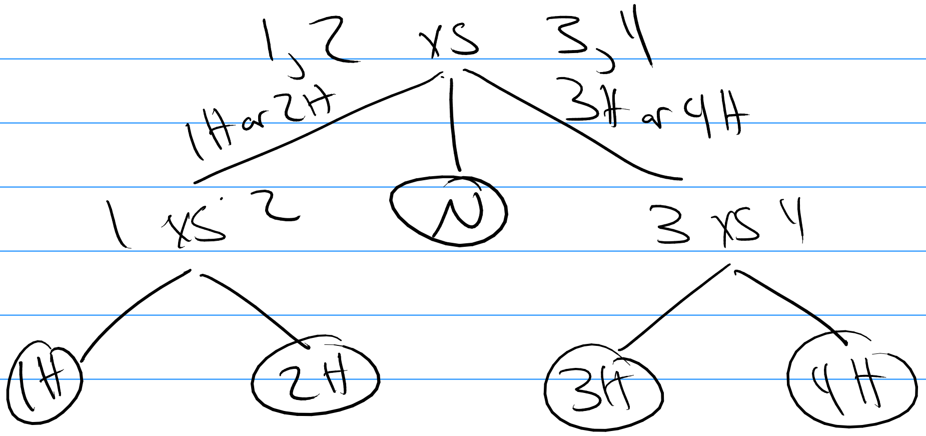
2 weighings

← 9 leaves



better!

best?

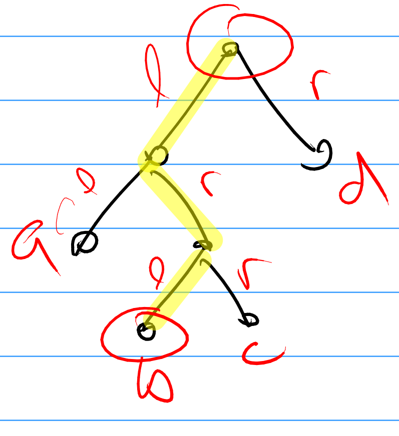


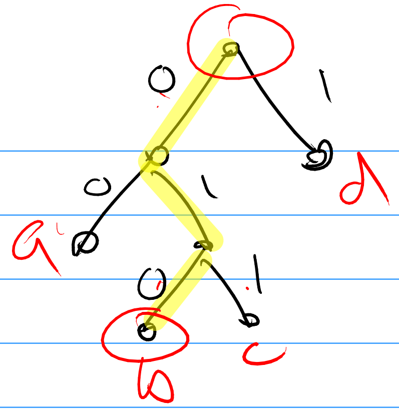
Prefix Codes | Huffman Code

Uniq. Simple path between any two vertices.

(root to leaf)

a :	l, l
b :	l, r, l
c :	l, r, r
d :	r



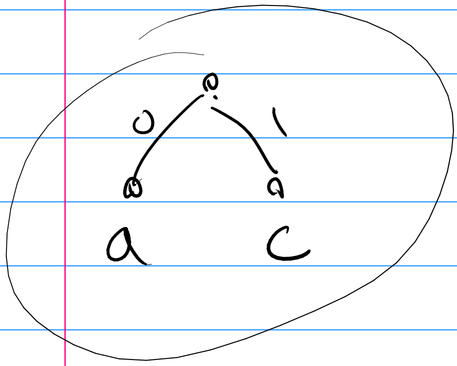


a: 00
b: 010
c: 011
d: 1

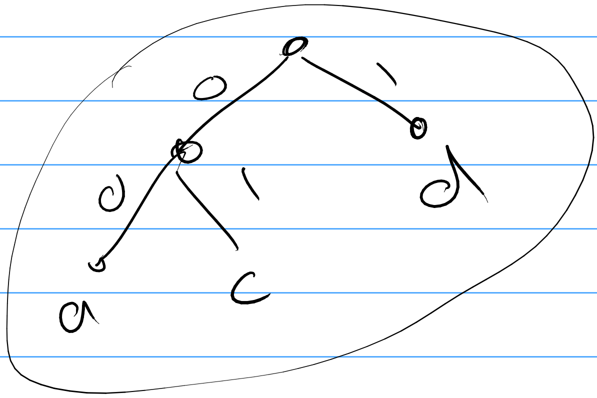
string: d d a a b c

bit string: $\underbrace{1}_{d}\underbrace{1}_{d}\underbrace{00}_{a}\underbrace{00}_{a}\underbrace{010}_{b}\underbrace{011}_{c}$

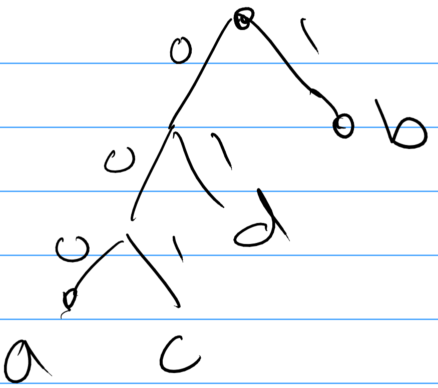
Huffman Code: a: 1% b: 90% c: 4% d: 5%



5%

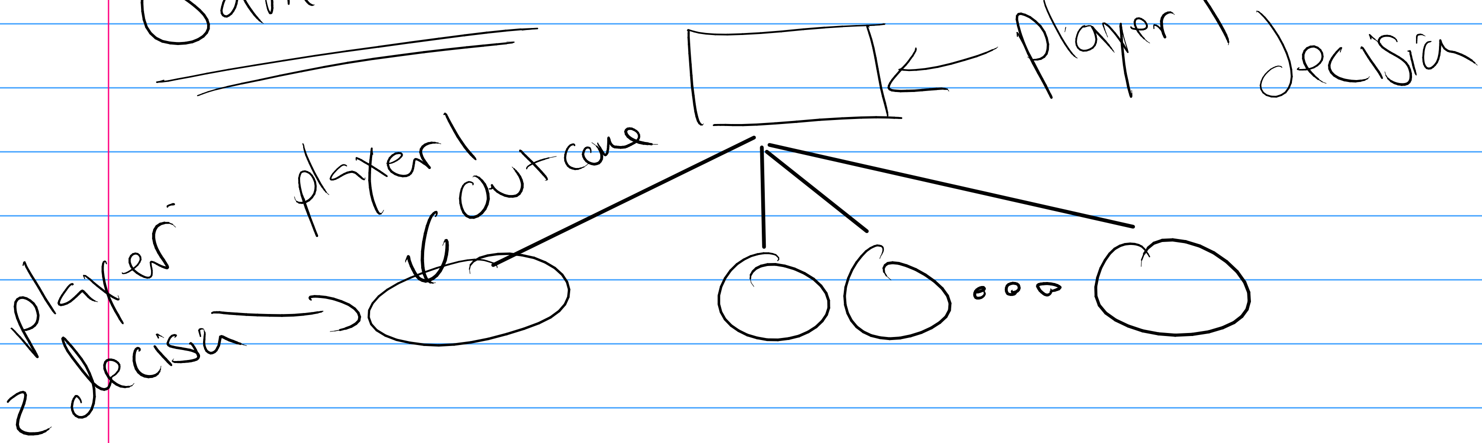


10%

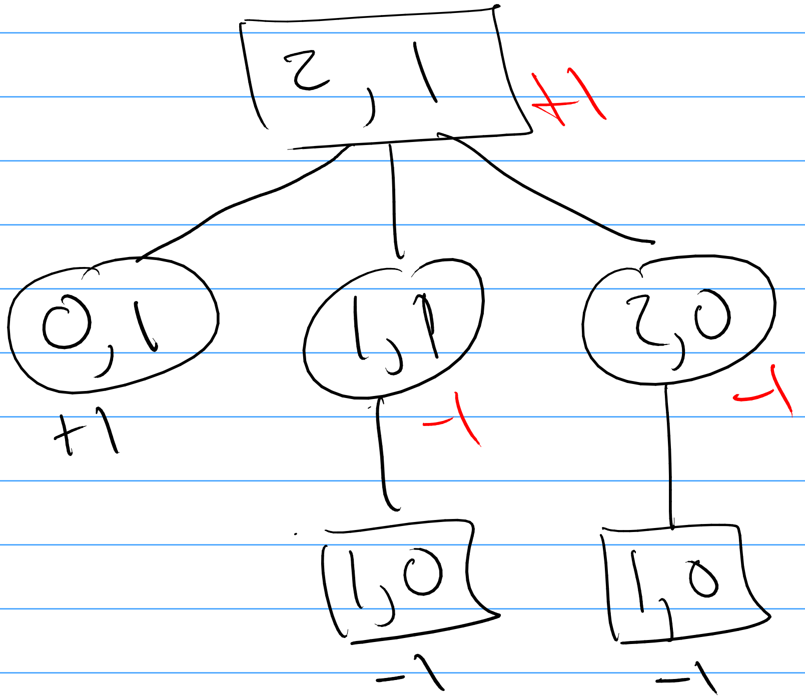


b: 1
d: 01
a: 000
c: 001

Game Trees.



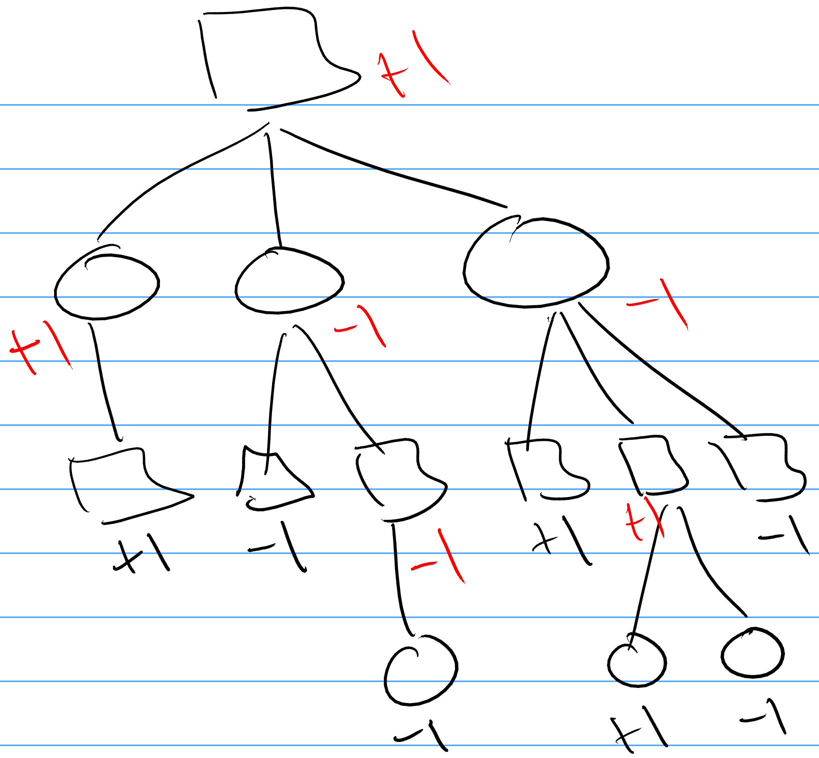
Nim : - take last stone = loss
- have to take something
- only work on one pile



label leaf with player 1's winnings.

Min/Max ← players play to win.

player two goes to minimize player 1
Player one goes to maximize player 1


+1
+1
-1
-1
+1
-1
-1
+1
+1
-1
-1
+1
-1