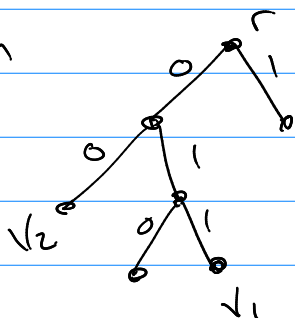


Math 322

Applications & Trees

unig. simple path from root to any vertex

① Binary Search



unig. simple path

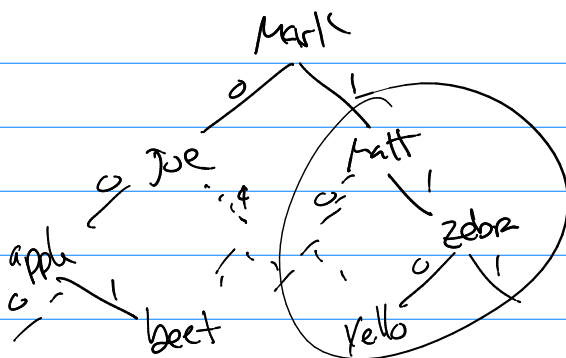
r to v1 011  
r to v2 000

Mark, Matt, Joe, Zebra, Yello, Apple, Beet

alphabetic

Joe, Mark, Matt, Zebra

binary:



Prefix coding / Huffman coding.

See M, a, r, K  
16bit, 16bit, 16bit, 16bit

16bit

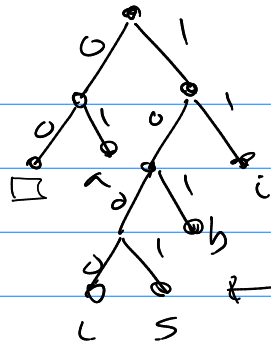
b<sub>1</sub>b<sub>2</sub>b<sub>3</sub>b<sub>4</sub> b<sub>5</sub>b<sub>6</sub>b<sub>7</sub>b<sub>8</sub> ... b<sub>16</sub>

a = 0000 0000 0000 0001  
b i

2<sup>16</sup>

□

low height  
place "common" symbols

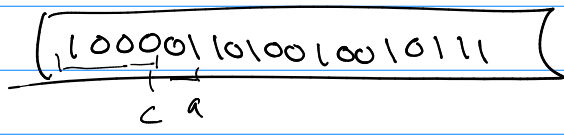


put symbols on leaves

- B: 00
- a: 01
- c: 1000
- s: 1001
- b: 101
- i: 11

greatest height  
place "rare" symbols

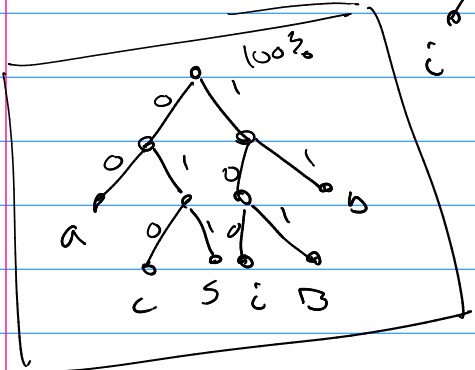
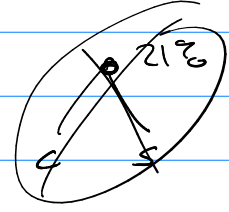
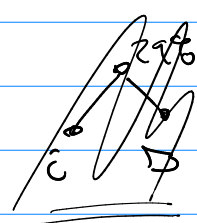
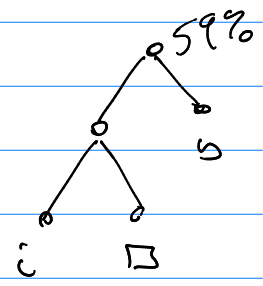
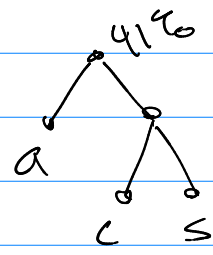
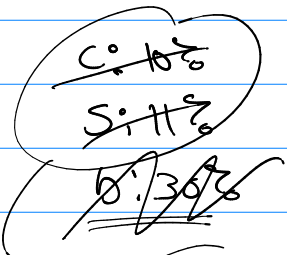
c a b s a i



Huffman: tech. to make the code.

language symbols: B, a, c, s, b

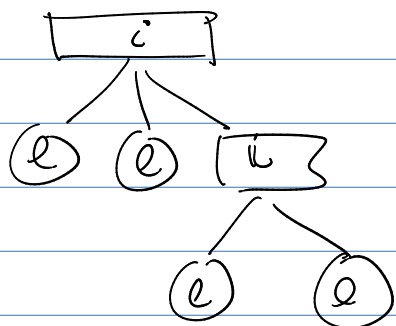
- ~~B: 15%~~
- ~~a: 20%~~
- ~~c: 14%~~



- a: 00
- c: 010
- s: 011
- i: 100
- B: 101
- b: 11

# Decision Tree

→ tree based process that has intermediate steps (step: decisions) that lead to a finite number of possible outcomes.



leaf = outcome

example: 100 objects to sort (linearly)  
how many outcomes?  $100!$

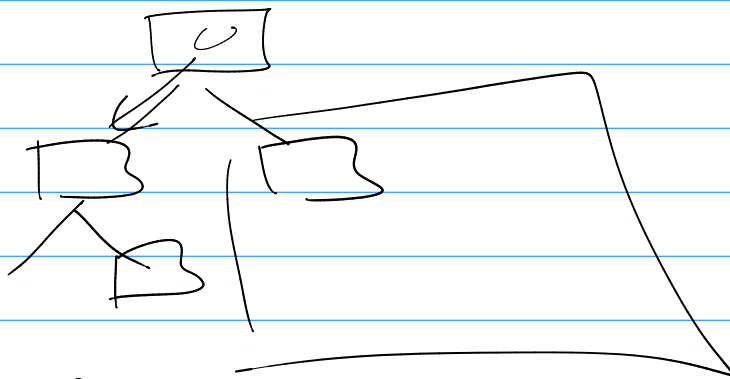
→ make a tree with  $100! = l$

$$h \geq \lceil \log_m l \rceil$$

(ex) binary tree  $m=2$

Depth:  $10^{150}$

$$\log 10^{150} = \underline{150 \log 10}$$



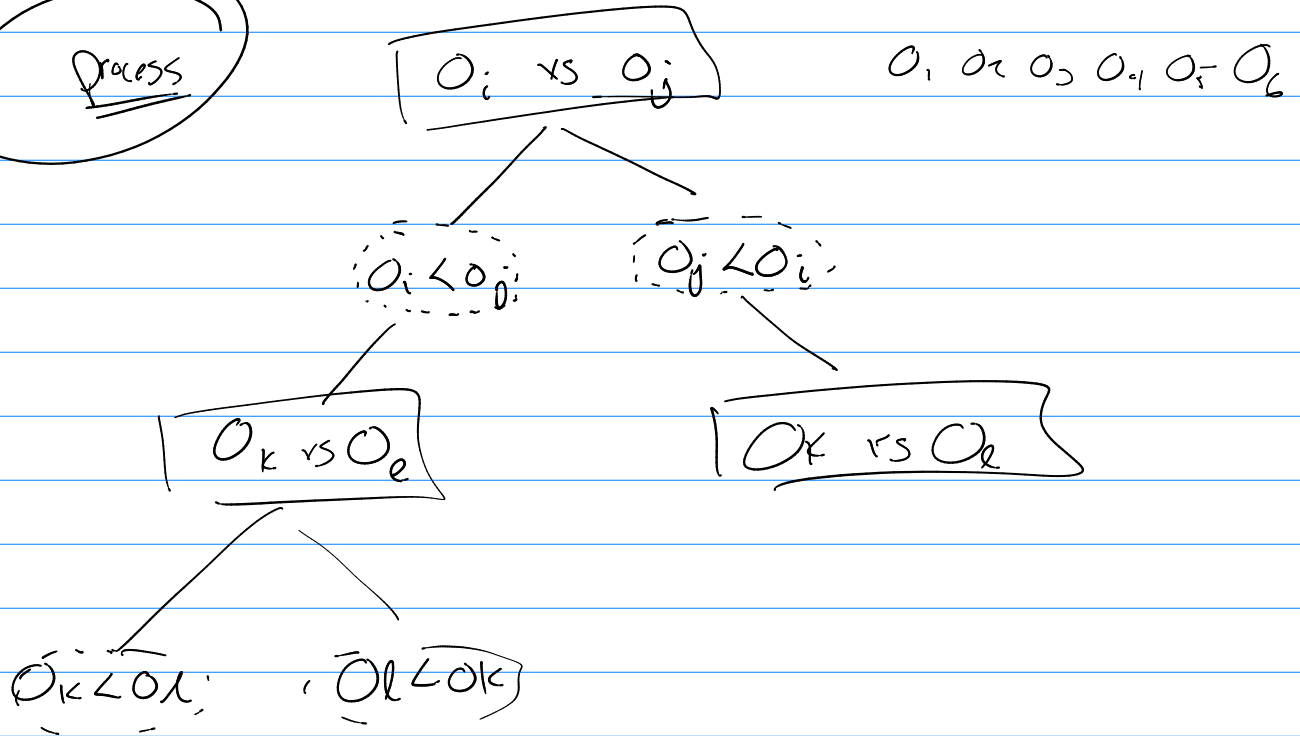
$h$  = max number of decisions to do

$h = \lceil \log_m l \rceil$  if we can make the tree full and balanced.

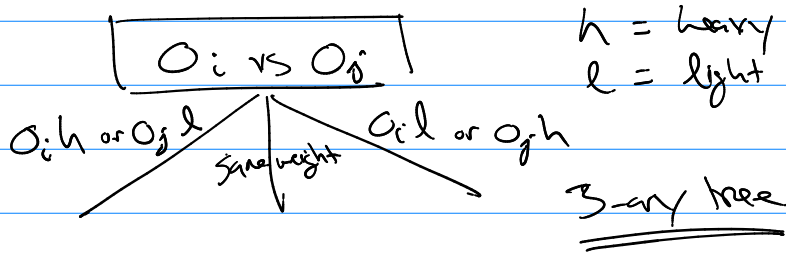
(ex) Sort 6 objects outcomes =  $l = 6! = 720$

binary tree ( $n=2$ )  $h \geq \lceil \log_2 720 \rceil = 10$

Process

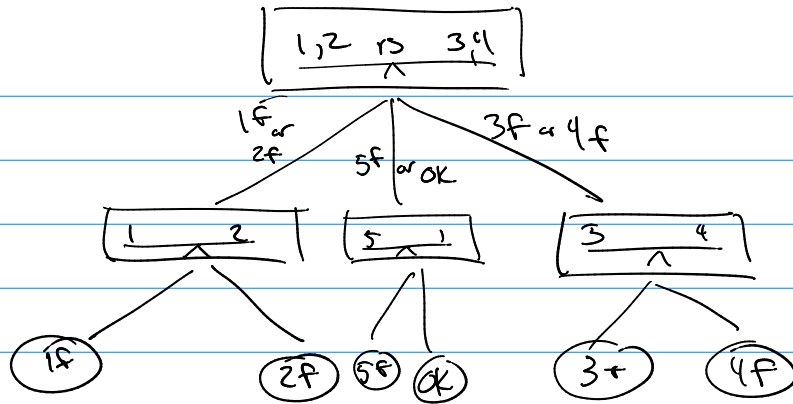


(ex) finding fake coins. use a balance scale



(ex) 5 coins and 1 could be fake. If it is, it is heavy.  
 $l = \text{outcomes: } 6 \text{ of sum.}$  (1), (2), (3), (4), (5)  
1 fake, 2 fake, 3f, 4f, 5f, ok

$h \geq \lceil \log_3 6 \rceil = 2$  ← best case is 2 weighings



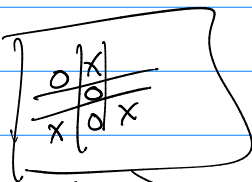
Game

internal vertex = game position

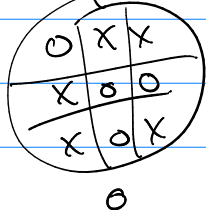
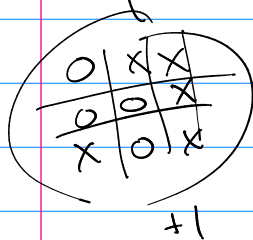
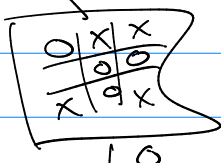
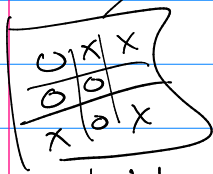
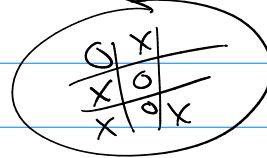
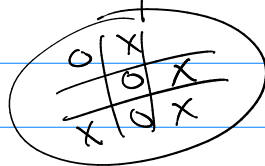
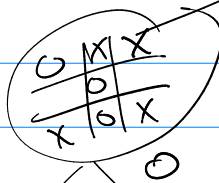
children = next positions

leaf = end of the game with table of player #1's payout.

ex

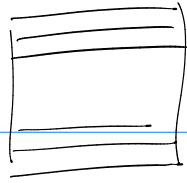


X's goes next (player 1)



Min/Max principle

Chess / Go



Chess

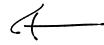
64 files, 8x8



20



20



400