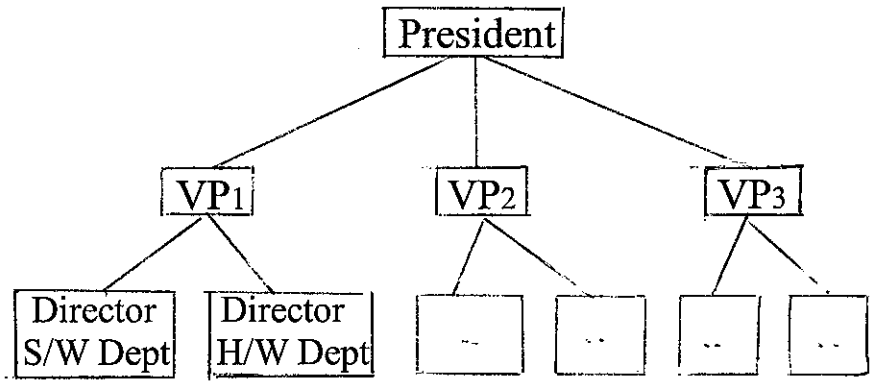
10 Trees

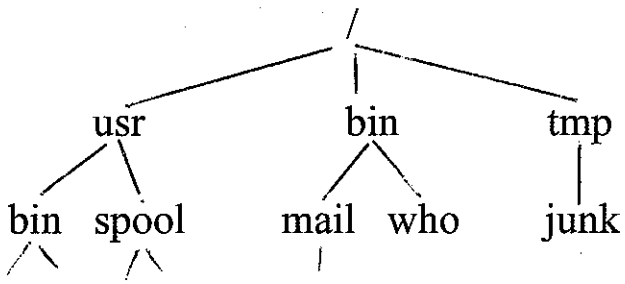
- connected, undirected graph with no simple cycle

Rooted tree

Ex. Organizational tree



Ex. Unix file system



Binary tree

(1) Binary Search Tree

Ex 1.

mathematics, physics, geography, zoology, meteorology, geography, psychology, and chemistry

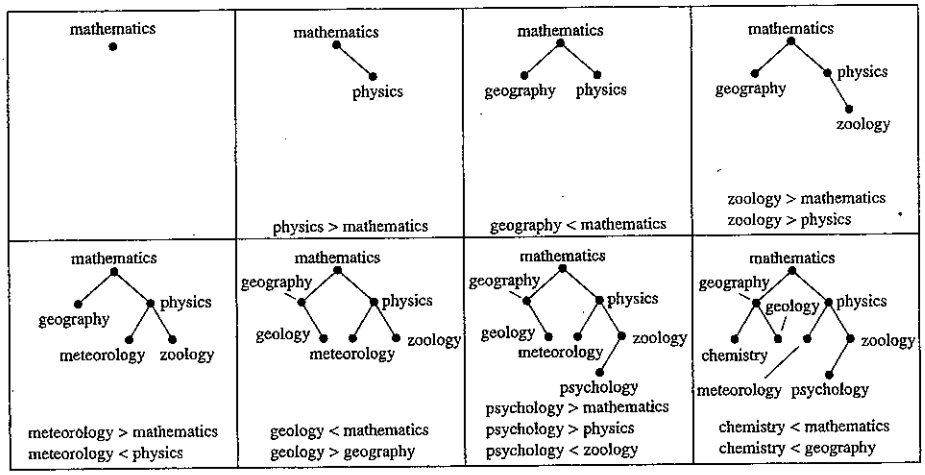


FIGURE 1 Constructing a Binary Search Tree.

(2) Decision Tree

Ex 3. Find a counterfeit coin among 8 coins. (minimum number of weighing)

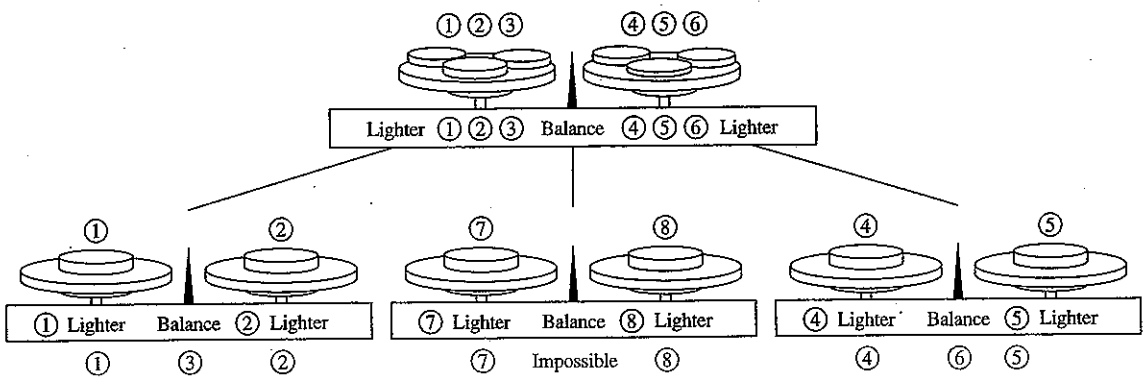


FIGURE 3 A Decision Tree for Locating a Counterfeit Coin. The counterfeit coin is shown in color below each final weighing.

Prefix Code

Problem. Assign bit strings to the English alphabet.

$2^k \geq 26 \rightarrow k = 5$

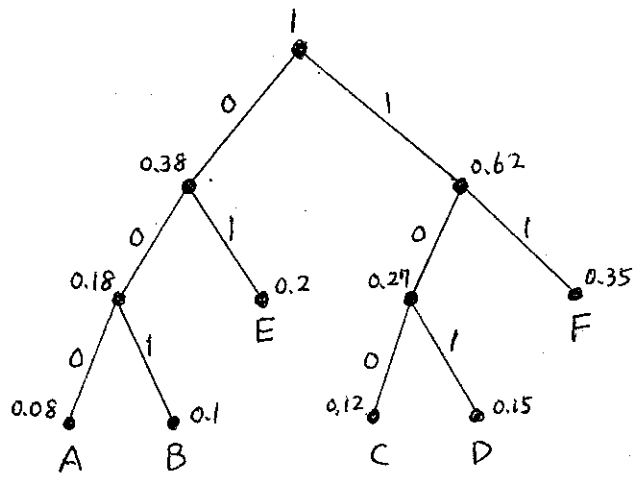
Solution. Five bits per letter → 5 x 26 = 130 bits

Huffman Code

Application – data compression
Greedy algorithm
Optimal binary prefix code.

Ex. 5

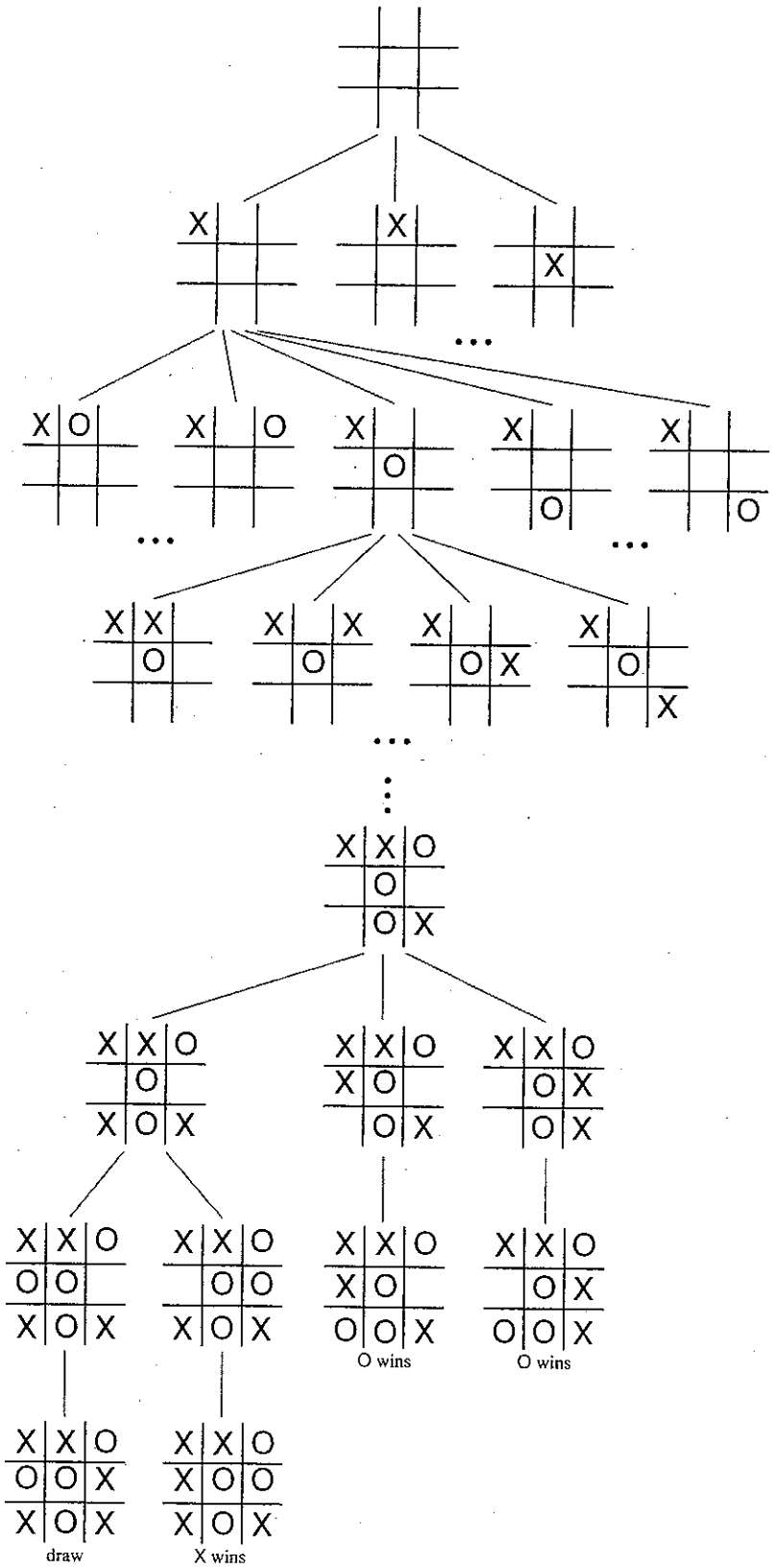
	frequency	code
A	0.08	000
B	0.1	001
C	0.12	100
D	0.15	101
E	0.2	01
F	0.35	11



Avg. length = (0.08)(3) + (0.1)(3) + (0.12)(3) + (0.15)(3) + (0.2)(2) + (0.35)(2)
= 2.45

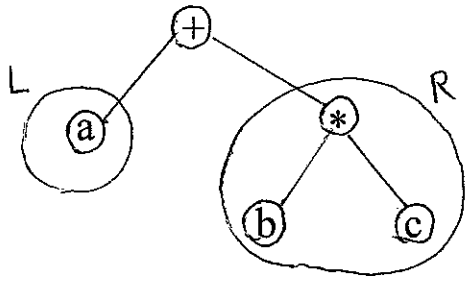
18.3% reduction

Game Tree for Tic-Tac-Toe.



10.3 Tree Traversal

Binary Tree Traversal



Rt, L, R → 3! different ordering

Rt	L	R	≡	preorder	(+ a * b c)
L	Rt	R	≡	inorder	(a + b * c)
L	R	Rt	≡	postorder	(a b c * +)

Note. Postorder
- Reverse Polish notation (Jan Łukasiewicz)
- Parenthesis free

Note. inorder + preorder → unique binary tree
inorder + postorder → unique binary tree

Note. infix → postfis
(a) Use stack (Dijkstra's algorithm)
(b) Use parenthesis

Ex. a + b * c

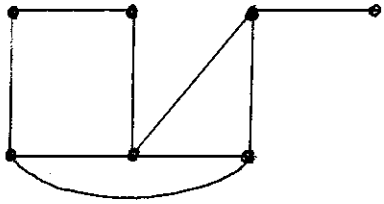
(a + (b * c))

a b c * +

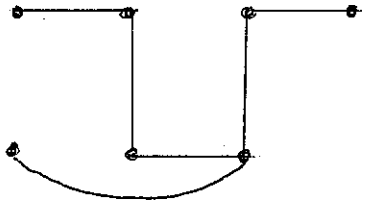
10.4 Spanning Tree

Def. A subgraph that is a tree containing every vertex of G

Ex.

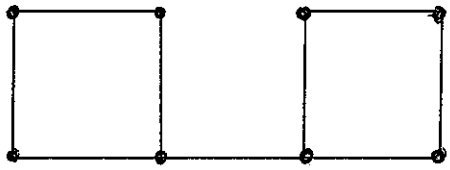


G



A spanning tree

Ex. How many spanning trees are there?

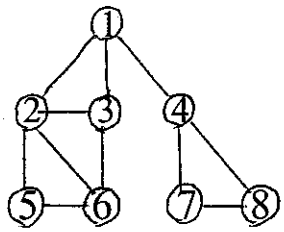


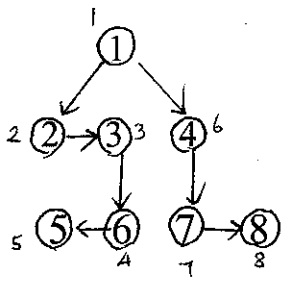
Depth-first Search (undirected)

```
procedure search (G)     // recursive //
begin
      procedure dfs(v)
        VISITED[v] := true
        for each node w adjacent to v do
          if not VISITED[w] then
            call dfs(w)
          endif
        endfor

    // main //
      for each v in V do
        VISITED[v]:= false
      endfor
      call dfs(s)   // s is the root
end
```

Ex.





DFS spanning tree

```
dfs(1)
  dfs(2)
    dfs(3)
      dfs(6)
        dfs(5)
  dfs(4)
    dfs(7)
      dfs(8)
```

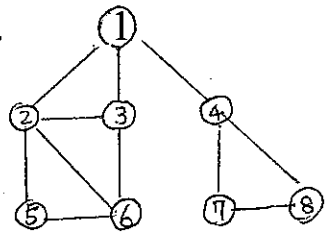
$T(n,m) = O(n + m)$

Breadth-first Search

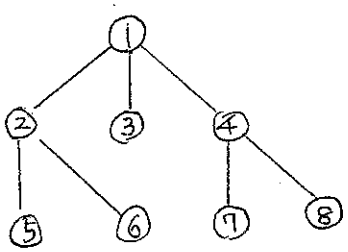
```
procedure search(G)        // non-recursive //
begin
      procedure bfs(v)
        Q:= ∅
        VISITED[v]:= true
        Q ← v    // insert v into Q //
         while Q not empty do
           delete w from Q
           for each w adjacent to v do
             if not VISITED[w] then
             begin
                 VISITED[w]:= true
                 Q ← Q ∪ w
             end
           endfor
         endwhile

     // main //
        for each v do VISITED[v]:= false
        call bfs(s)    // s is the root
  end {search}
```

Example.



Node visited	Q
1	2, 3, 4
2	3, 4, 5, 6
3	4, 5, 6
4	5, 6, 7, 8
5	6, 7, 8
6	7, 8
7	8
8	–

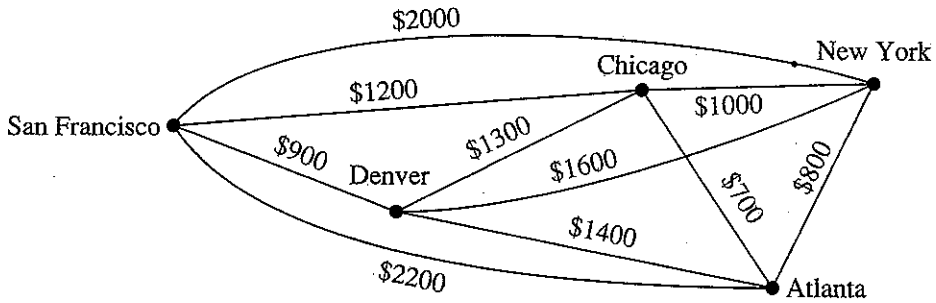


BFS tree

$T(n,m) = O(n+m)$

Note shortest path tree when each edge length is 1.

10.5 Minimum Spanning Tree

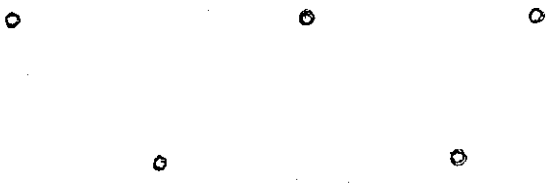


Monthly lease

- Prim's (1957): nearest-neighbor

$T_W =$

- Kruskal's (1956): lightest-edge



$T_W =$