Important Problems

- Traveling Salesman Problem

"Hamilton Circuit"

a

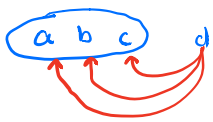
All permutations of immediate nodes

⇒ Exhaustive search ⇒

- knapsack Problem

Optimize the value

→ Best combo of items



exhaustive

(DFS)

- goes to far nodes first

Vertex

1)
2)
3)
4)
5)

Depth First Search Forest:

-> use a stack to trace algorithm

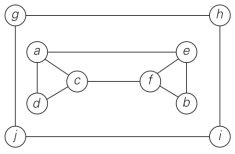
(BFS)

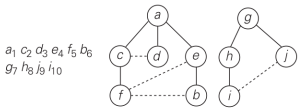
- goes to near nodes First

Pick a,

X

Cross links





not connected

Cycle edges!

Adjacency matrix

Adjacency lists

Vertices edges

n(n-1)

ALGORITHM *DFS*(G)

//Implements a depth-first search traversal of a given graph
//Input: Graph $G = \langle V, E \rangle$
//Output: Graph G with its vertices marked with consecutive integers
// in the order they are first encountered by the DFS traversal
mark each vertex in V with 0 as a mark of being "unvisited"
$count \leftarrow 0$
for each vertex v in V **do**
    if v is marked with 0
        dfs(v)

dfs(v)
//visits recursively all the unvisited vertices connected to vertex v
//by a path and numbers them in the order they are encountered
//via global variable *count*
$count \leftarrow count + 1$; mark v with *count*
for each vertex w in V adjacent to v **do**
    if w is marked with 0
        dfs(w)

ALGORITHM *BFS*(G)

//Implements a breadth-first search traversal of a given graph
//Input: Graph $G = \langle V, E \rangle$
//Output: Graph G with its vertices marked with consecutive integers
// in the order they are visited by the BFS traversal
mark each vertex in V with 0 as a mark of being "unvisited"
$count \leftarrow 0$
for each vertex v in V **do**
    if v is marked with 0
        bfs(v)

bfs(v)
//visits all the unvisited vertices connected to vertex v
//by a path and numbers them in the order they are visited
//via global variable *count*
$count \leftarrow count + 1$; mark v with *count* and initialize a queue with v
while the queue is not empty **do**
    for each vertex w in V adjacent to the front vertex **do**
        if w is marked with 0
            $count \leftarrow count + 1$; mark w with *count*
            add w to the queue
    remove the front vertex from the queue

4.1, 4.2, 4.4

* Look at Shell sort

- Decrease and Conquer

- Two Types:

  - 1

  / 2

Sorted

Find location