Breadth-First Search (BFS)

Graph Search

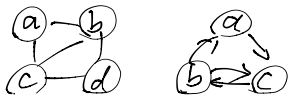
"explore graph"

graph $G = (V, E)$ V: set of vertices

E: set of edges

directed → $e = \{v, w\}$ unordered pairs

directed → $e = (v, w)$ ordered pairs

e.g.: 

undirected directed

$E = \{\{a,b\}, \{b,c\}, \{c,d\}, \ldots\}$ $E = \{(a,c), (b,c), (c,b), (b,a)\}$

Application

- Web crawling : google search
- Social networking : facebook find friend
- Network broadcast
- garbage collection : modern programming languages
- model checking
- checking mathematical conjecture
- Solving puzzles & games

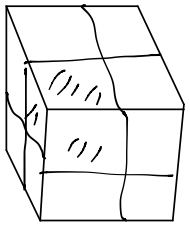
- Pocket Cube : 2 × 2 × 2
  - Configuration : graph
  - vertex for each possible state of cube

    # Vertices = $8! \cdot 3^8$

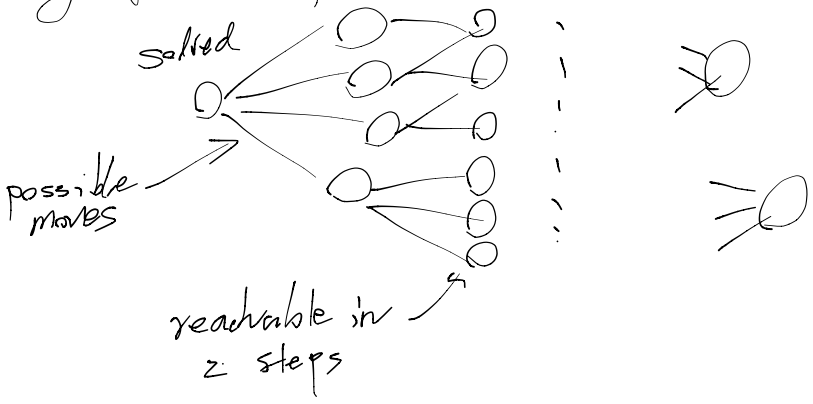
    $= 264,539,520$

    # Unique & reachable:

    # Vertices / 24 / 3



  - edge for each possible move



diameter of graph = 11

13 × 3 × 3 ⟹ 20

2 × 2 × 2 : 11

3 × 3 × 3 : 20

n × n × n : $\Theta\left(\frac{n^2}{\lg n}\right)$

Graph representation

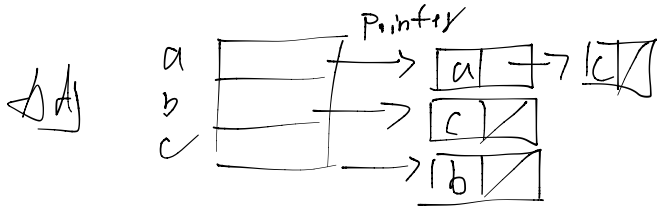
- Adjacency lists
  - Array Adj of size |V| linked lists
  - For each vertex $u \in V$, Adj[u] stores u's neighbors

$Adj[u] = \{ v \in V \mid (u,v) \in E \}$

directed

$Adj[b] = \{ a, c \}$ $Adj[c] = \{ b \}$

$Adj[a] = \{ c \}$



Adj can be realized:

- array
- hash table

— (4) $O(|V|+|E|)$ (directed graph)

- Object-oriented

  v.neighbors = $Adj[v]$

- Implicit representation
  - $Adj(u)$ is a function
  - or v.neighbor() is a method

☐ Breadth-first Search (BFS)

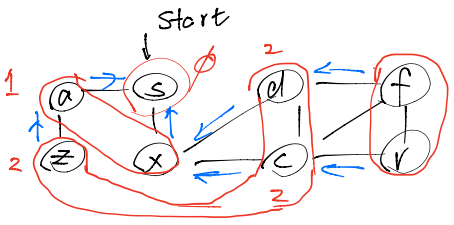
Simplest algorithm in graph search

- Visit all nodes, reachable from given node $s \in V$
- $O(V+E)$ time

- look at nodes reachable in 0 move, 1 move, 2 moves, ...
- careful to avoid duplicates (cycle)

```
BFS(s, Adj):
    level = {s: 0}
    parent = {s: None}
    i = 1
    frontier = [s]          ← level i-1
    while frontier:
        next = []           ← level i
        for u in frontier:
            for v in Adj[u]:
                if v not in level:
                    level[v] = i
                    parent[v] = u
                    next.append(v)
        frontier = next
        i += 1
```

- e.g.:



level i frontiers → parent pointers (all leads to s)

- Shortest Path

— $v \leftarrow parent[v]$

$\leftarrow parent[parent[v]]$

$\leftarrow \cdots\cdots$

$\leftarrow s$

is a shortest path from s to v

$=$

of length $level[v]$

running time $\sum_{v \in V} |Adj[v]| = \begin{cases} 2|E|, & \text{undirected} \\ |E|, & \text{directed} \end{cases}$

handshaking lemma