Algorithms on Graphs

A graph is a pair $G = (V, E)$ consisting of a set of vertices V and a set of edges $E \subseteq V \times V$.

Graphs are widely used to represent e.g.

- Molecules
- All kinds of networks (reaction networks, regulatory networks, computer networks, social networks ...)

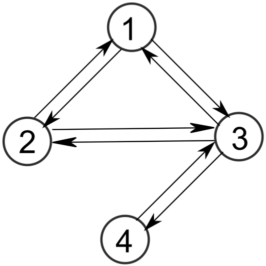
- can be directed or undirected
- can carry labels/weights on edges and/or vertices

- In contrast to trees, many problems on graphs are computationally hard

Representing Graphs

Most commonly represented as

- Nodes for vertices and pointers representing links
- Adjacency matrix
- Adjacency list (if the matrix is sparse)



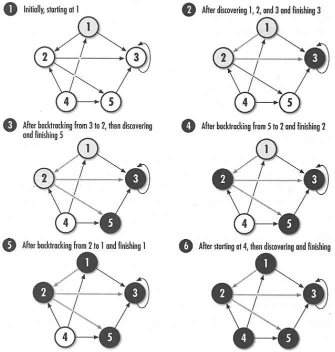
$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{aligned} A[1] &= \{2, 3\} \\ A[2] &= \{1, 3\} \\ A[3] &= \{1, 2, 4\} \\ A[4] &= \{3\} \end{aligned}$$

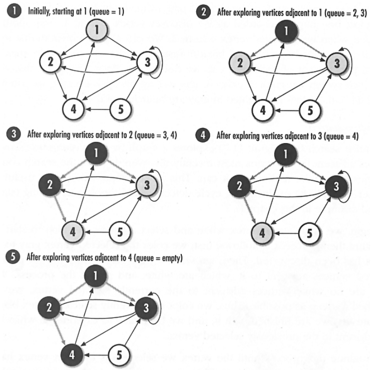
Traversing a Graph

There are two important search methods from which many important graph algorithms are derived.

Depth-first-search



Breadth-first-search



Depth First Search

Algorithm 14: DFS(G)

Data: Depth First Search starting a v

```
foreach v ∈ V do
    color[v] = WHITE;
end
foreach v ∈ V do
    if color[v] = WHITE then
        DFSvisit(G, v);
    end
end

DFSvisit(G, v)
color(v) = GRAY;
foreach u ∈ Adj[v] do
    If color[u] = WHITE  DFSvisit(G, u);
end
color(v) = BLACK;
```

Breadth First Search

Algorithm 15: BFS(G, s)

Data: Breadth First Search starting a s, Queue Q
foreach $u \in V - \{s\}$ **do**
    $color[u] =$ WHITE;
end
while $Q \neq \emptyset$ **do**
    $u = head[Q]$;
    foreach $v \in Adj[u]\}$ **do**
        if $color[v] = WHITE$ **then**
            $color[v] =$ GRAY;
            ENQEUE(Q, v);
        end
        DEQUEUE(Q);
        $color[u] =$ BLACK;
    end
end

Dijkstra’s algorithm

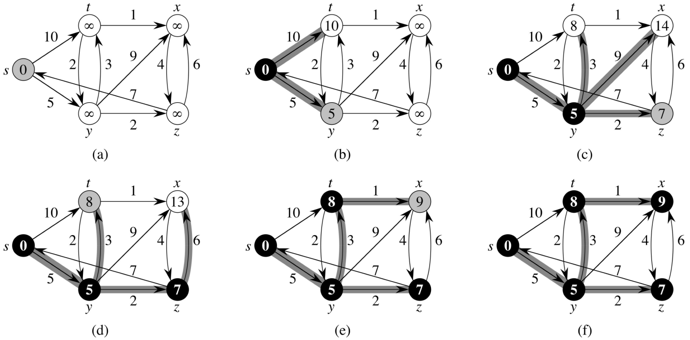
In a weighted graph find the shortest path between two nodes.
In fact, computes distance to all nodes from some start point.

Dijkstra's algorithm

Algorithm 16: Dijkstra(G, s)

```
add all vertices to Q;
d[s] = 0; prev[s] = NULL;
foreach v ∈ V − {s} do
    d[v] = ∞;
    prev[v] = NULL;
end
while Q ≠ ∅ do
    u = vertex with minimal d[u];
    remove u from Q;
    foreach v ∈ Adj[u]} do
        t = d[u] + len(u, v);
        if t <d[v] then
            d[v] = t;
            prev[v] = u;
        end
    end
end
```

Dijkstra's algorithm



Time complexity:

- original version $\mathcal{O}(|V|^2)$
- with min-priority queue $\mathcal{O}(|E| + |V| \log |V|)$

Graph Invariants

- A graph invariant is a graph property independent of its representation

  e.g. planar, connected, bipartite

- Often a numerical value $\rightarrow$ graph index

  e.g. diameter, number of cycles

- In chemical graph theory called topological index or connectivity index

  e.g. Wiener index, Hosoya index, Balaban index

Example: Wiener Index

$$W = \frac{1}{2} \sum_{ij} d_{ij}$$

$$\begin{aligned} \text{n-Butane:} \quad W &= 3 \cdot 1 + 2 \cdot 2 + 1 \cdot 3 = 10 \\ \text{IsoButane:} \quad W &= 3 \cdot 1 + 3 \cdot 2 = 9 \end{aligned}$$

Example: Wiener Index

$$W = \frac{1}{2}\sum_{ij} d_{ij}$$

$$\begin{aligned}
\text{n-Butane:} \quad W &= 3 \cdot 1 + 2 \cdot 2 + 1 \cdot 3 = 10 \\
\text{IsoButane:} \quad W &= 3 \cdot 1 + 3 \cdot 2 = 9
\end{aligned}$$

- Oldest index, introduced in 1947
- More compact structures have smaller Wiener index
- Boiling points of alkanes correlate with W
- Origin of chemical descriptor based QSAR