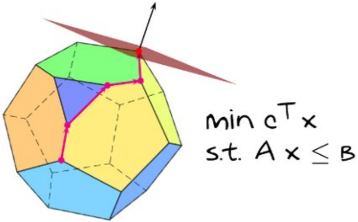
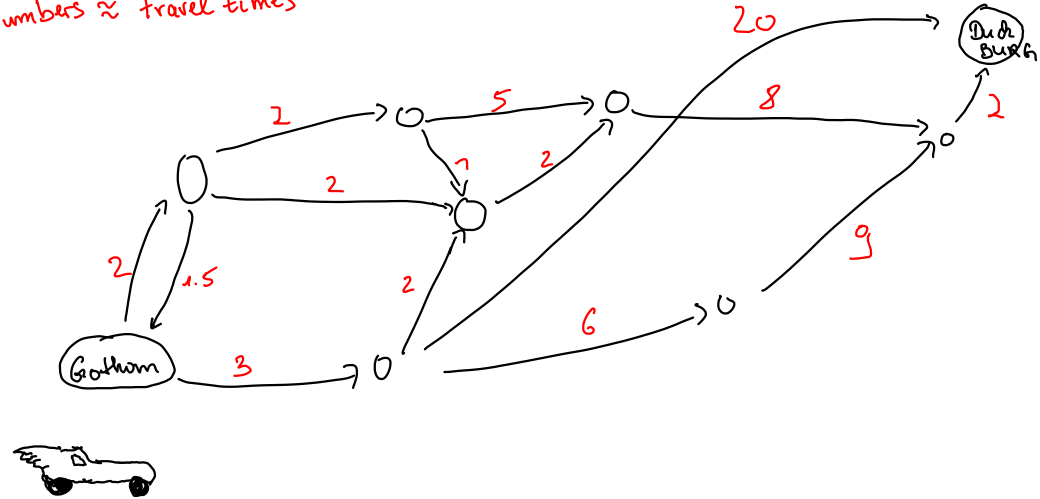
Paths, Cycles and Flows

- Directed graphs
- Shortest (unweighted) paths
- Breadth-First-Search



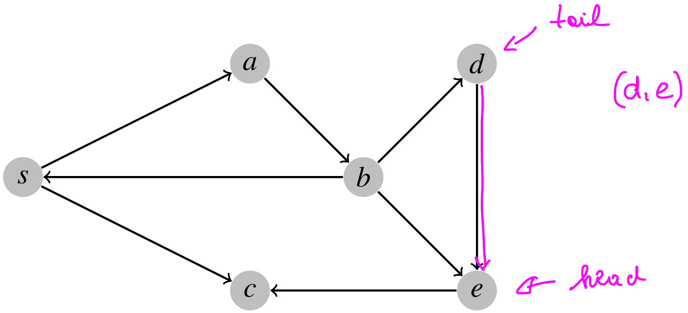
Motivation

Numbers ≈ travel times



Directed graphs

A *directed graph* is a tuple $D = (V, A)$, where V is a finite set of *vertices* or *nodes* and $A \subseteq (V \times V)$ is the set of *arcs* or *directed edges* of G.

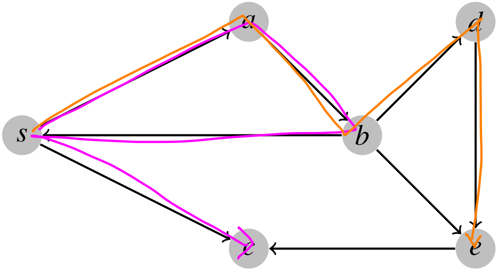


We denote a directed edge by its defining tuple $(u, v) \in A$. The nodes u and v are called *tail* and *head* of (u, v) respectively.

Walks and paths

A *walk* in a directed graph $D = (V, A)$ is a sequence $v_0, \ldots, v_k$, where $(v_i, v_{i+1}) \in A$ for each $i = 0, \ldots, k-1$.

A walk is a *path* if the $v_0, \ldots, v_k$ are distinct. The *length* of the path $v_0, \ldots, v_k$ is k.



s, a, b, s, c

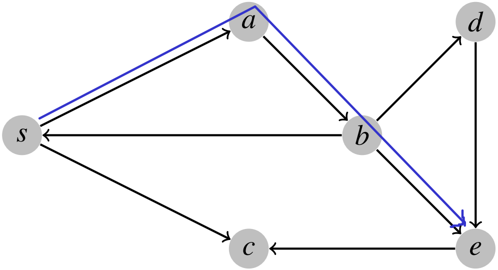
s, a, b, d, e : PATH

length = 4

Unweighted distance

The *distance* $d(s,t)$ between two nodes $s,t \in V$ is the smallest $k \in \mathbb{N}_0$ such that there exists a path $s = v_0, \ldots, v_k = t$. (Possibly ∞).

$d(s,t)$ is the length of the *shortest path* connecting s and t.



Quiz:

$d(s,e) = \boxed{3}$

Quiz

What is the largest possible length of a path a directed graph $D = (V, A)$ with $|V| = n$?

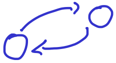
- n
- $n - 1$
- $n^2 - 1$

$v_1 \to v_2 \to \cdots \to v_n$

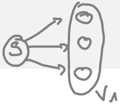
Which of the following are upper bounds for the number of directed paths of length $n - 1$ in directed graph with n nodes?

- $n!$
- 2^n
- n

PATH SPECIFIED BY Sequence $v_1, v_2, \dots, v_n$ $n!$



Distance labels



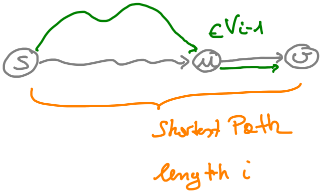
For $i \in \mathbb{N}_0$, $V_i \subseteq V$ denotes the set of vertices that have distance i from s. Notice that $V_0 = \{s\}$.

Proposition

For $i = 1, \ldots, n-1$, the set V_i is equal to the set of vertices $v \in V \setminus (V_0 \cup \cdots \cup V_{i-1})$ such that there exists an arc $(u, v) \in A$ with $u \in V_{i-1}$.

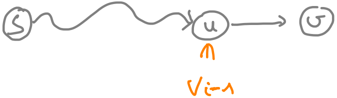
Proof:

„$\subseteq$“ $v \in V_i$

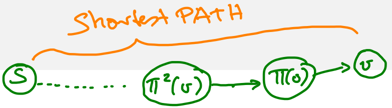


„$\supseteq$“ $v \in V_j$, $j \geq i$

$j \leq i$ $\Rightarrow j = i$



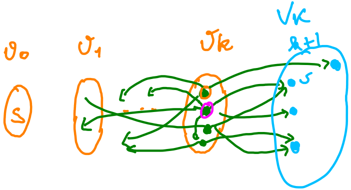
Breadth-First-Search

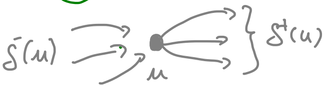


Alg. maintains two arrays

$$\begin{array}{c} 0 \quad \infty \quad\quad \infty \\ D[v_1 = s, v_2, \dots, v_n] \\ \pi[v_1 = s, v_2, \dots, v_n] \\ 0 \quad 0 \quad\quad 0 \end{array}$$

and a *queue* Q. $= [S]$





while $Q \neq \emptyset$

  $u := head(Q)$

  for each $(u, v) \in \delta^+(u)$

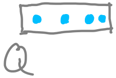
    if $(D[v] = \infty)$

      $\pi[v] := u$

      $D[v] := D[u] + 1$

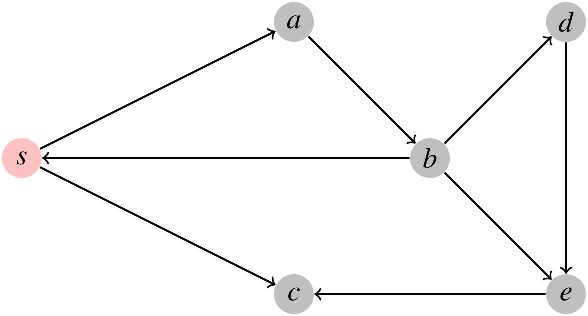
      $enqueue(Q, v)$

  $dequeue(Q)$



Example

while $Q \neq \emptyset$
$\quad u := head(Q)$
$\quad$ **for** each $(u, v) \in \delta^+(u)$
$\quad\quad$ **if** $(D[v] = \infty)$
$\quad\quad\quad \pi[v] := u$
$\quad\quad\quad D[v] := D[u] + 1$
$\quad\quad\quad enqueue(Q, v)$
$\quad dequeue(Q)$



$Q = [s]$

Example

while $Q \neq \emptyset$

$u := head(Q)$

for each $(u, v) \in \delta^+(u)$

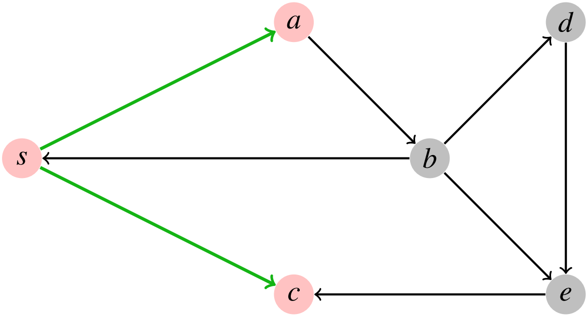
if $(D[v] = \infty)$

$\pi[v] := u$

$D[v] := D[u] + 1$

$enqueue(Q, v)$

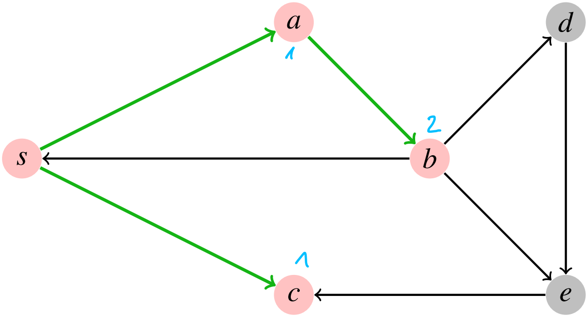
$dequeue(Q)$



$Q = [a, c]$

Example

while $Q \neq \emptyset$
$u := head(Q)$
for each $(u, v) \in \delta^+(u)$
if $(D[v] = \infty)$
$\pi[v] := u$
$D[v] := D[u] + 1$
$enqueue(Q, v)$
$dequeue(Q)$



$Q = [c, b]$

$Q = \boxed{b}$

Example

while $Q \neq \emptyset$

$\quad u := head(Q)$

$\quad$ **for** each $(u, v) \in \delta^+(u)$

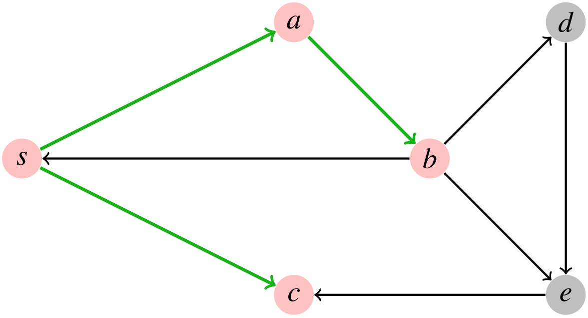
$\quad\quad$ **if** $(D[v] = \infty)$

$\quad\quad\quad \pi[v] := u$

$\quad\quad\quad D[v] := D[u] + 1$

$\quad\quad\quad enqueue(Q, v)$

$\quad dequeue(Q)$



$Q = [b]$

Example

while $Q \neq \emptyset$

$u := head(Q)$

for each $(u, v) \in \delta^+(u)$

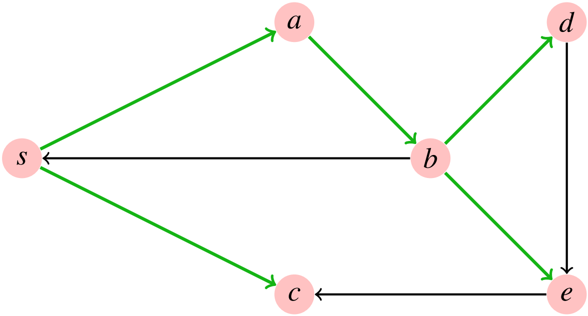
if $(D[v] = \infty)$

$\pi[v] := u$

$D[v] := D[u] + 1$

$enqueue(Q, v)$

$dequeue(Q)$



$Q = [d, e]$

Example

while $Q \neq \emptyset$

$u := head(Q)$

for each $(u, v) \in \delta^+(u)$

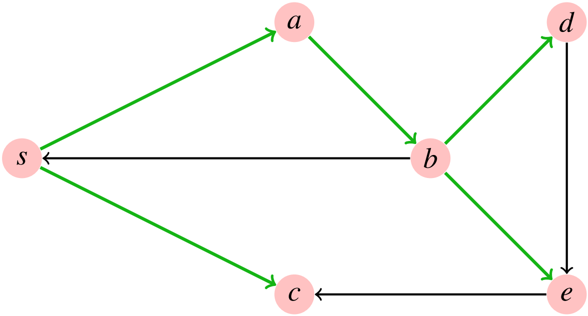
if $(D[v] = \infty)$

$\pi[v] := u$

$D[v] := D[u] + 1$

$enqueue(Q, v)$

$dequeue(Q)$



$Q = [e]$

Example

while $Q \neq \emptyset$

$\quad u := head(Q)$

$\quad$ **for** each $(u, v) \in \delta^+(u)$

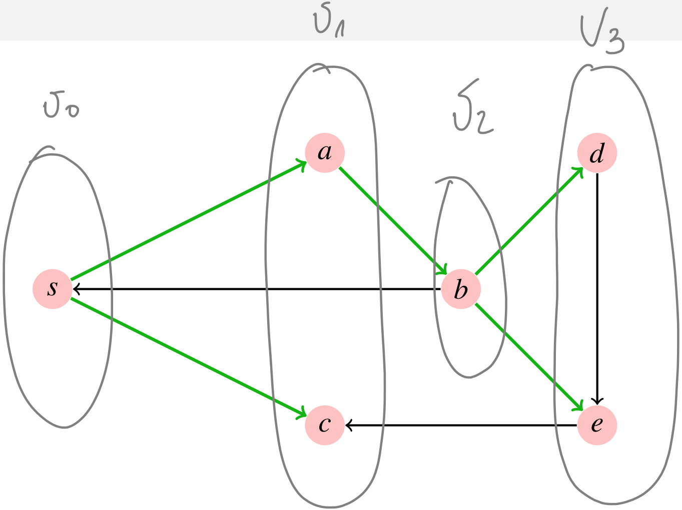
$\quad\quad$ **if** $(D[v] = \infty)$

$\quad\quad\quad \pi[v] := u$

$\quad\quad\quad D[v] := D[u] + 1$

$\quad\quad\quad enqueue(Q, v)$

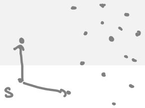
$\quad dequeue(Q)$



$Q = []$

Cheat: Ignore init. of ARRAYS.

With this initialization $O(|V| + |A|)$



Theorem

The Breadth-First-Search algorithm runs in time $O(|A|)$. It is thus a *linear time* algorithm.

Iteration u: At most $c_1 \cdot |\delta^+(u)| + c_2$ elementary operations.

while $Q \neq \emptyset$
    $u := head(Q)$
    for each $v \in \delta^+(u)$
        if $(D[v] = \infty)$
            $\pi[v] := u$
            $D[v] := D[u] + 1$
            $enqueue(Q, v)$
    $dequeue(Q)$

$|\delta^+(u)|$

$c_1 \cdot |\delta^+(u)| + c_2$

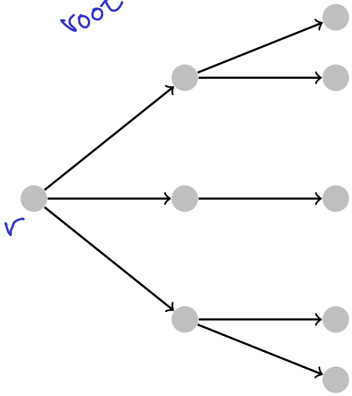
$$\sum_{\substack{u \in V \\ \text{reachable from } s}} c_1 \cdot |\delta^+(u)| + c_2 \leq c_1 \cdot |A| + |V_{\text{reach. from } s}|$$

$= O(|A|)$ elementary op. $\square$

Directed trees

A *directed tree* is a directed graph $T = (V, A)$ with $|A| = |V| - 1$ and there exists a node $r \in T$ such that there exists a path from r to all other nodes of T.

root



A shortest path tree

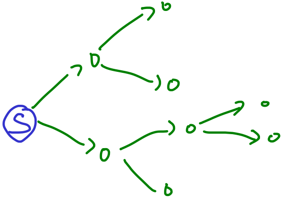
Lemma

Consider the arrays D and π after the termination of the breadth-first-search algorithm. The graph $T = (V', A')$ with $V' = \{v \in V : D[v] < \infty\}$ and $A' = \{(\pi(v), v) : 1 \leqslant D[v] < \infty\}$ is a tree.

Proof.

s

- Clearly, $|A'| = |V'| - 1$.
- For any $i \in \{1, \ldots, n-1\}$, by backtracking the π-labels from any $v \in V_i$, we will eventually reach s.



Shortest path tree

□