HONORS ALGORITHMS | 2024-01-24 | 1

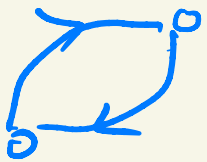
Digraph · directed graph $G=(V,E)$

edges: → $E \subseteq V \times V$

set of order pairs

we permit loops (v,v)

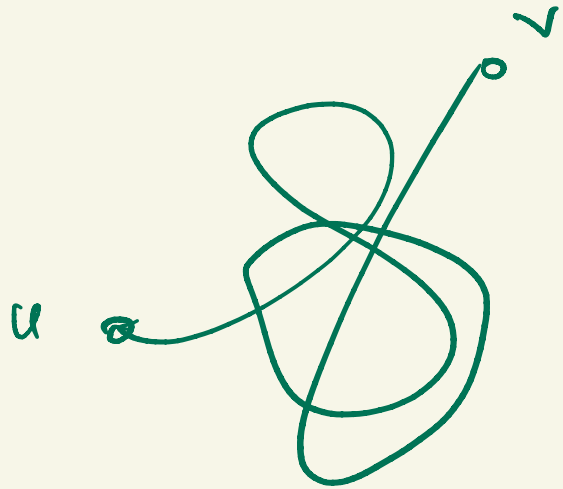
Standard representation of a graph as a digraph

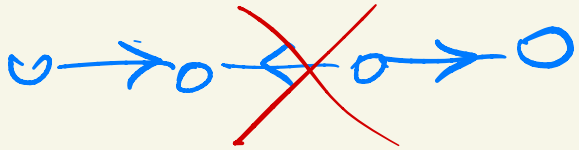


(directed)
walk of length k for u to v

$u = w_0 - w_1 - \dots - w_k = v$

$w_{i-1} \to w_i$





subdigraph $H = (W, F)$ $G = (V, E)$

$H \subseteq G$ if $W \subseteq V$, $F \subseteq E$

set of out-neighbors $N_G^+(u)$

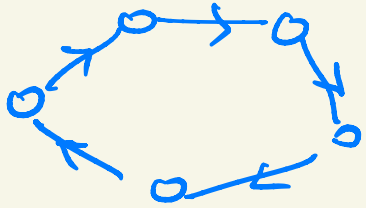
$N_G^-(u)$

$\deg^+(u) = |N_G^+(u)|$ out-degree

$\deg^-(v) = |N^-(v)|$ in-degree

directed path of length $n-1$

$\vec{P}_4$



$\vec{C}_5$

$\vec{C}_1$ ↻

$\vec{C}_2$ ↻↻

under. cycles have length ≥ 3

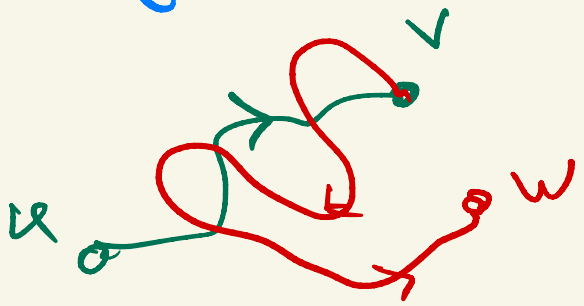
A path in a digraph G is a subdigraph that is [isomorphic to] a path

DEF v is accessible from u $\exists$ $u \dashrightarrow v$ path

DEF v is **accessible** from u $\exists\ u \to \cdots \to v$ **path**

LEMMA v is $-''-$ u $\Leftrightarrow \exists u \to \cdots \to v$ **walk**

$\therefore$ accessibility is a transitive relation



Lemma a shortest walk $u \to \cdots \to v$ is a path

EX given a $u \to \cdots \to v$ walk

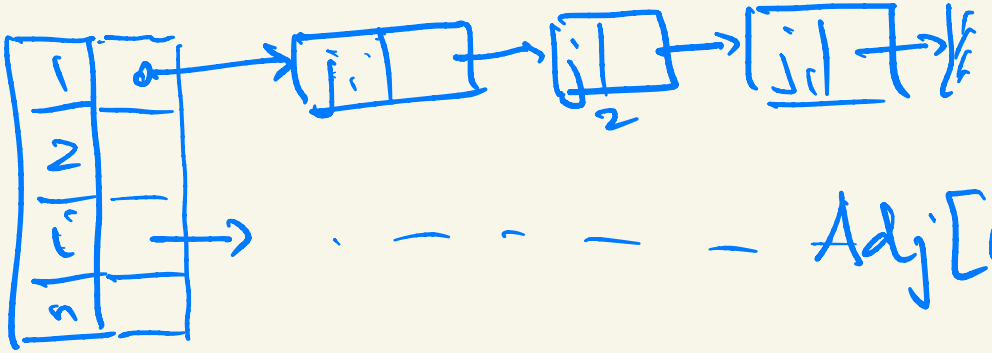
find $u \to \cdots \to v$ path in linear time

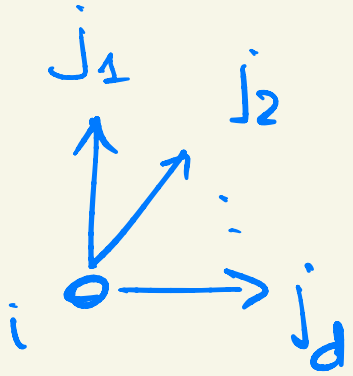
Decide $u \to \cdots \to v$ **accessibility** $-''-$

REPRESENTING a digraph:

ARRAY OF ADJACENCY LISTS

$V = [n]$





"random access"
given an address (link)
at unit cost we can
go to the address

adjacency list
of vertex i

name of a vertex : token $\in [n]$

UNIT COST MODEL:

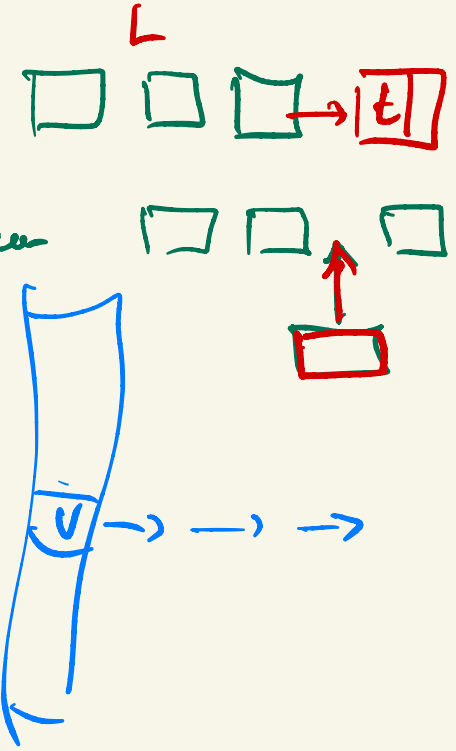
operations with tokens: unit cost

following links
- next
- first
- arithmetic $\pm$

APPEND (L, t)

DELETE current item

INSERT after " "



SINGLE PASS ALGORITHM scheme

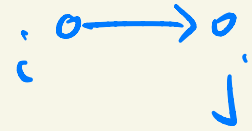
```
for   v ∈ V
    for   w ∈ Adj[v]
        do  something   (remember v
                         we know: edge v→w)
    endfor
endfor
```

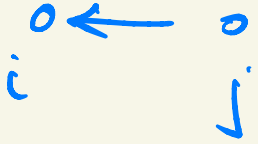
Cost $O(n+m)$

$n = |V|$

$m = |E|$

G^{tr} reverse (transpose) of G

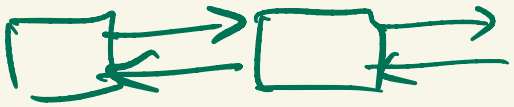


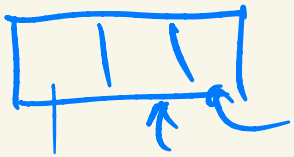


(adj. matrix transposed)

Ex Given an adj. list. rep. of G
find — " — of G^{tr} | single pass

doubly linked list : in lin time, add backward links





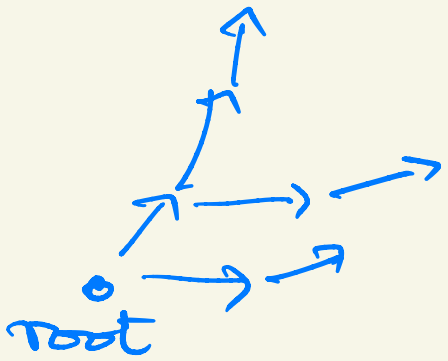
add link to last

lin time [monotone adj. list: neighbors come in increasing order, remove multiplicities (repetitions)]

---

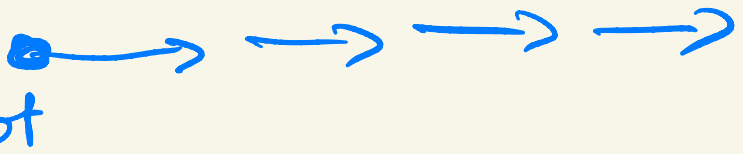
conversion between array and linked list lin time (single pass)

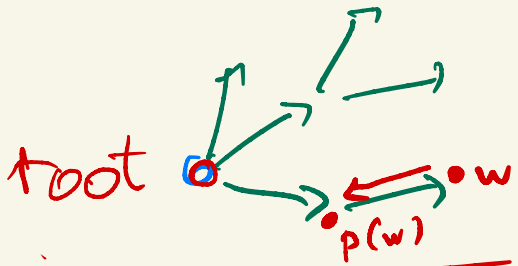
SINGLE SOURCE SHORTEST PATHS problem



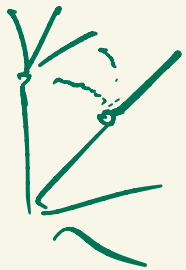
listing < shortest root → vertex paths

could be quadratic





"parent" link toward root



from every accessible vertex

|

from root

ALGORITHM: BREADTH-First Search HANDOUT

FIFO List first-in-first-out