

Discussion Week 2

1] Warm Up

2] TopSort

3] S-t numpaths

1] T or F?

a) Running DFS on a directed-acyclic graph (DAG) results in no back-edges. back-edge means cycle, DAG has no cycles.

b) In a directed G with 2 SCCs, it is always possible to add one edge to make G' 1 SCC.



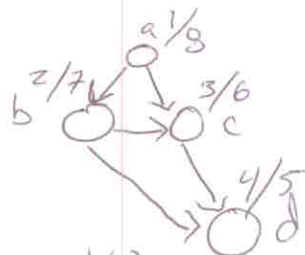
can't add 1 edge to make 1 SCC.

2] TopSort: Input: DAG

Output: Assign vertices a number $n(v) \in \{1, \dots, |V|\}$ such that $\forall e = (u, v) \in E, n(u) < n(v)$.

"Linearize" graph.

• All DAGs can be linearized w/DFS!



decreasing post = a, b, c, d

$n(v) = 1, 2, 3, 4$

• Only edges for which $post(u) < post(v)$ are back edges... DAG has none!

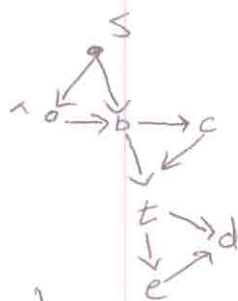
• All edges go from left to right.

[3] S-t Numpaths

Input: DAG $G=(V,E)$; vertices s, t

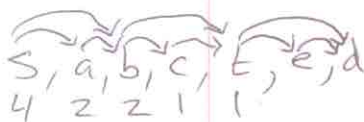
Output: Number of different paths from s to t .

Ex:



s, b, t
 s, a, b, t
 s, b, c, t
 s, a, b, c, t
= 4 paths.

Idea: TopSort



• Let $\text{numpaths}[v] = \# \text{ of paths from } v \text{ to } t$.

• 0 if v is to right of t in TopSort

$$\text{numpaths}[e] = \text{numpaths}[d] = 0$$

• $\text{numpaths}[t] = 1$

$$\text{numpaths}[v] = \sum_{e=(v,x) \in E} \text{numpaths}[x]$$

$$\text{e.g. } \text{numpaths}(c) = \text{numpaths}(t) = 1$$

$$\begin{aligned} \text{numpaths}(b) &= \text{numpaths}(c) + \text{numpaths}(t) \\ &= 2 \end{aligned}$$

Alg: NumPaths(G, s, t)

- 1) TopSort(G) (uses DFS + post numbers), let $n(v)$ be their Top Sort order
- 2) Set $\text{numpaths}(v) = 0 \quad \forall v \text{ w/ } n(v) > n(t)$
- 3) Set $\text{numpaths}(t) = 1$
- 4) For $i = n(t) - 1$ to 1 decreasing
 - Let x have $n(x) = i$
 - Set $\text{numpaths}(x) = \sum_{e=(x,z) \in E} \text{numpaths}(z)$
- 5) Return $\text{numpaths}(s)$.

Correctness

Claim: ~~G~~ G has k ^{distinct} paths from s to t iff

$\text{Numpaths}(G, s, t)$ returns k .

~~" $\Rightarrow$ " Assume G has k distinct paths from s to t .
Then $\text{Numpaths}(G, s, t) = k$.~~

SubClaim: $\text{numpaths}(x) = \#$ of paths from x to t .

Pf: By induction.

Base: Let x ~~be~~ ~~neighbor~~ have $n(x)+1 = n(t)$.

If there's an edge (x, t) $\text{numpaths}(x) = 1$, otherwise

$\text{numpaths}(x) = 0$. ✓

For x with $n(x) > n(t)$, $\text{numpaths}(x) = 0$. ✓

Induction: Assume $\text{numpaths}(x) = \#$ of paths

from x to t , $\forall x$ w/ $n(x) \geq n(t)$.

Show $\text{numpaths}(k)$ is correct.

Only way to get to t is through k 's neighbors which all ~~are~~ have TopSort #

bigger than k . So $\text{numpaths}(k) = \sum_{(k, z) \in E} \text{numpaths}(z)$

back to claim.

" $\Rightarrow$ " Assume G has k distinct paths from s to t ,
since we return $\text{numpaths}(s)$, k will be returned
by subclaim.

" $\Leftarrow$ " If Numpaths returns k , then there must be k
paths to t from s by subclaim.