

Recitation 9

DFS

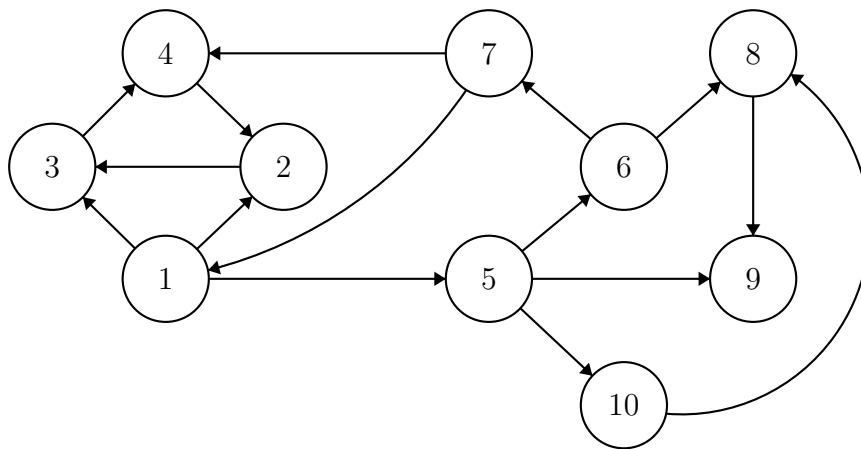
9.1 Announcements

- *BridgeLab* has been released, and is due **Thursday night**. It's worth 125 points.
- *ShortLab* will be released on **Thursday**.

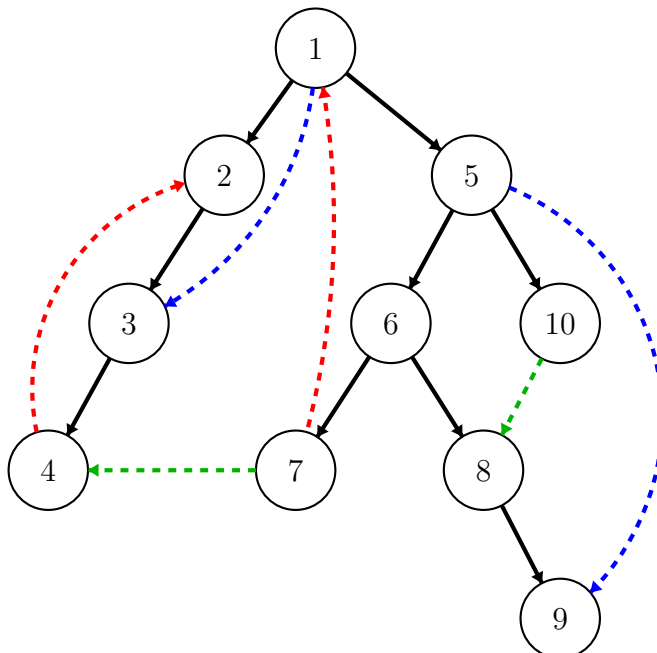
9.2 DFS Trees and Numberings

Task 9.1. Starting at vertex 1, execute DFS on the following graph, visiting vertices in increasing order. Trace the process by doing each of the following.

1. Draw the resulting DFS tree. Draw tree edges as solid lines, and include non tree edges in your drawing as dashed lines.
2. Classify each non tree edge as one of **forward**, **back**, or **cross**.
3. Label each vertex with its discovery and finish times.



In the following diagram, back edges are red, forward edges are blue, and cross edges are green.



Vertex	Discovery	Finish
1	0	19
2	1	6
3	2	5
4	3	4
5	7	18
6	8	15
7	9	10
8	11	14
9	12	13
10	16	17

Task 9.2. Suppose DFS is run on a directed graph, and consider some edge (x, y) . Use the discovery and finish times of x and y to classify this edge.

Write $d[v]$ and $f[v]$ for the discovery and finish time of v , respectively.

Numbering	Possible Edge Type
$d[x] < d[y] < f[y] < f[x]$	tree, forward
$d[y] < d[x] < f[x] < f[y]$	back
$d[y] < f[y] < d[x] < f[x]$	cross

9.2.1 Higher-Order DFS

Recall the following code from the textbook:

Algorithm 9.3. Directed, generalized DFS.

```

1  directedDFS (revisit, discover, finish) ( $G, \Sigma_0, s$ ) =
2    let
3      DFS  $p$  ( $(X, \Sigma), v$ ) =
4        if ( $v \in X$ ) then ( $X, revisit(\Sigma, v, p)$ ) else
5          let
6             $\Sigma' = discover(\Sigma, v, p)$ 
7             $X' = X \cup \{v\}$ 
8             $(X'', \Sigma'') = iterate(DFS\ v)(X', \Sigma')(N_G^+(v))$ 
9             $\Sigma''' = finish(\Sigma', \Sigma'', v, p)$ 
10         in
11         ( $X'', \Sigma'''$ )
12       end
13     in
14     DFS  $s$  ( $(\{\}, \Sigma_0), s$ )
15   end

```

Task 9.4. Define Σ_0 , $revisit$, $discover$, and $finish$ to calculate DFS numberings.

Algorithm 9.5. Time-stamping with generalized directed DFS.

```

1   $\Sigma_0 = (\{\}, \{\}, 0)$ 
2   $revisit(\Sigma, \_, \_) = \Sigma$ 
3   $discover((D, F, c), v, \_) = (D \cup \{v \mapsto c\}, F, c + 1)$ 
4   $finish(\_, (D, F, c), v, \_) = (D, F \cup \{v \mapsto c\}, c + 1)$ 

```

Task 9.6. *Modify the given generalized DFS code to work with undirected graphs.*

(Hint: We only want to traverse each edge once! Try implementing undirected cycle detection with the above algorithm and see where it fails.)

The problem with running the above code on an undirected graph is that every *every child will revisit its parent in the DFS tree, creating m back edges*. Hence, when attempting undirected cycle detection, every edge will be considered a cycle. We can fix this problem by omitting the parent from the neighbors of each child.

Algorithm 9.7. *Undirected, generalized DFS.*

```

1 undirectedDFS (revisit, discover, finish) ( $G, \Sigma_0, s$ ) =
2   let
3      $\text{DFS } p ((X, \Sigma), v) =$ 
4       if ( $v \in X$ ) then ( $X, \text{revisit } (\Sigma, v, p)$ ) else
5         let
6            $\Sigma' = \text{discover } (\Sigma, v, p)$ 
7            $X' = X \cup \{v\}$ 
8            $(X'', \Sigma'') = \text{iterate } (\text{DFS } v) (X', \Sigma') (\underline{N_G^+(v) \setminus p})$ 
9            $\Sigma''' = \text{finish } (\Sigma', \Sigma'', v, p)$ 
10        in
11           $(X'', \Sigma''')$ 
12        end
13    in
14       $\text{DFS } s ((\{\}, \Sigma_0), s)$ 
15    end
```