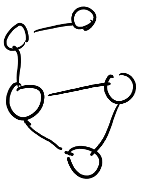
Graphs (DFS)

Tuesday, December 1, 2020 5:00 PM

Reminder: lab 8 is due next Monday.
HW 9 " " " Thursday.
Final is next Thursday (Dec 10th)
lab 7 → Thursday 11:59pm

Depth First Search (DFS):

applications:

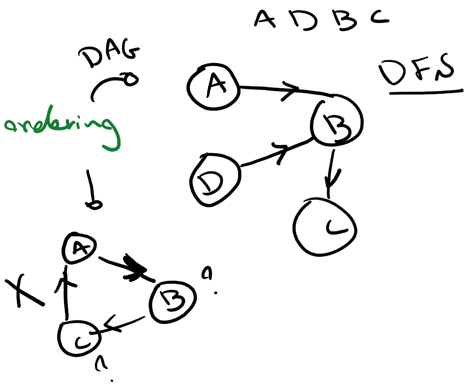


① To see what vertices are reachable from one initial vertex.

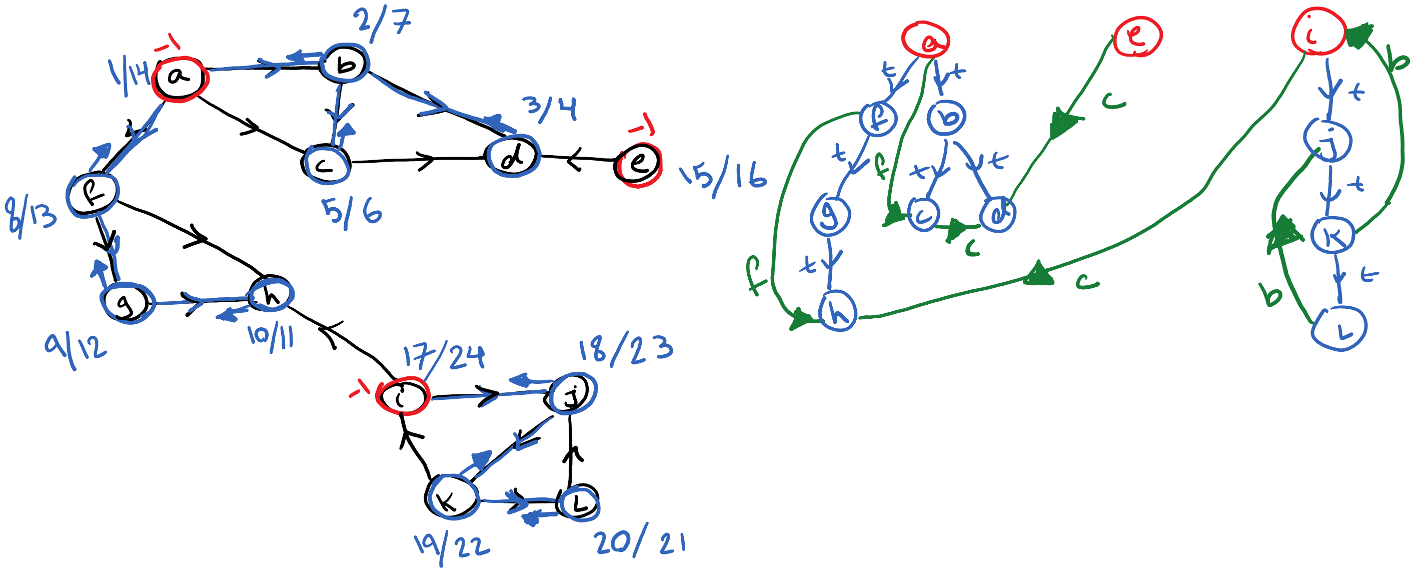
② Cycle detection

③ Topological sorting/ordering

④ Solving mazes



Example: Run DFS

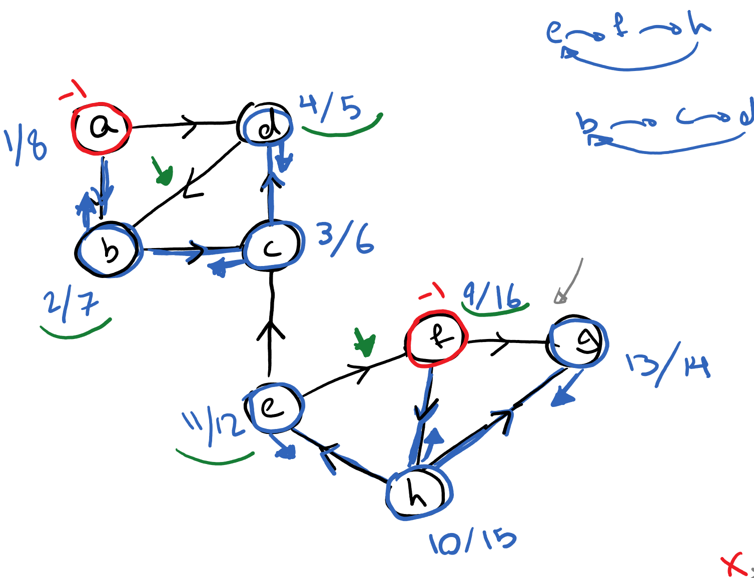


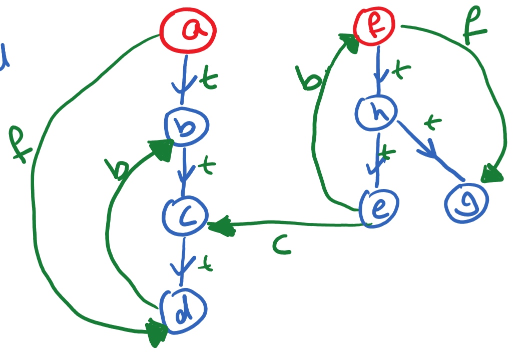
Edge Classification

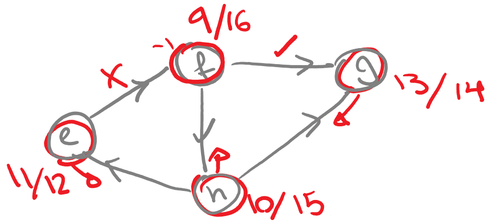
Edge Classification

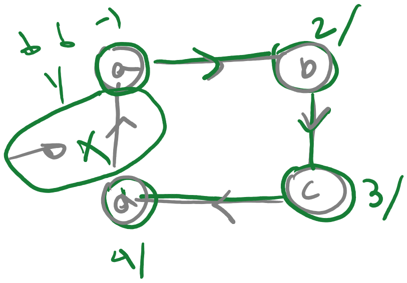
- * Tree edge: (have parent pointer) The ones we used in DFS to visit a new vertex.
- * Forward edge: Connects a node to its descendant.
- * Backward edge: ancestor. (creates a cycle)
- * Cross edge: All the other edges

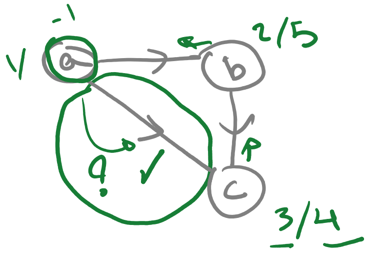
Example: Run DFS and then label the edges.

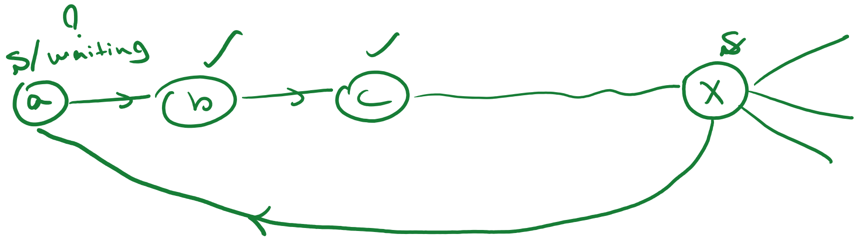




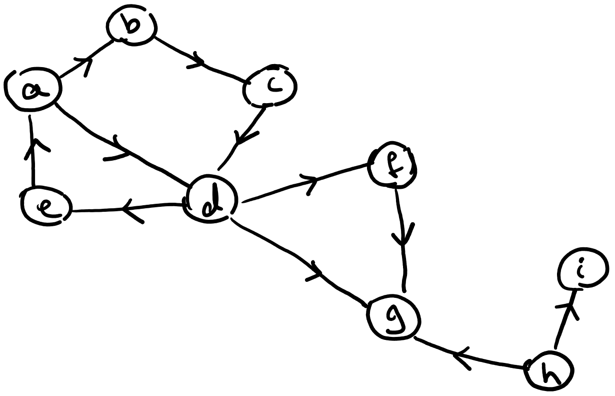








Example: Run DFS and label the edges

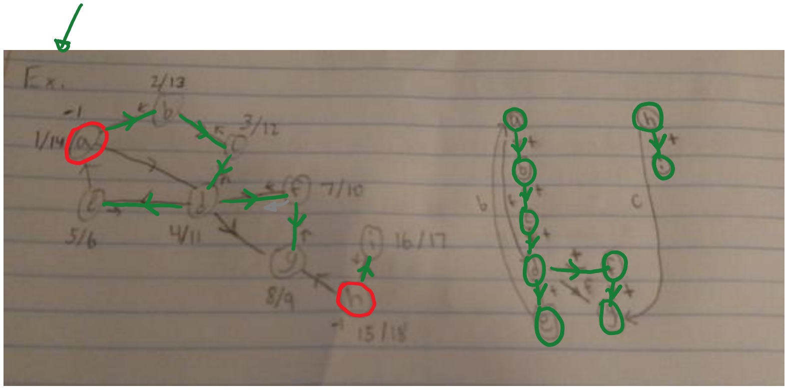


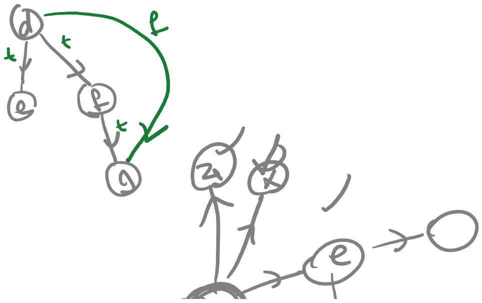
```
DFS(V)
    for v_i ∈ V
        → if ~~~~ // visited?
            → set parent pointer
              DFS_visit(v_i)
end

→ DFS_visit(v_i) ← BFS
                         v_i.s

                         v_i.e
  look for cycles
```

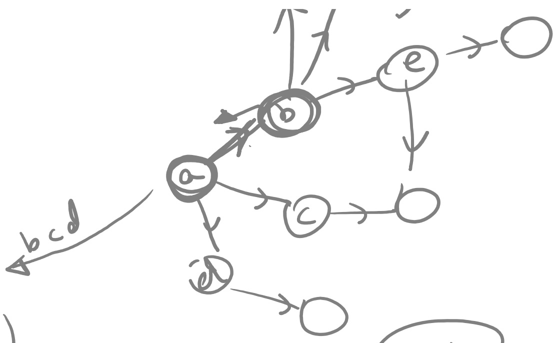
Matthew :)





```
DFS_visit(v_i)
    start timer

    for u_j ∈ v_i.adj
        if (check sth)
            set parent pointer for u_j
        else if (check for cycle)
                // print
        end
    end
    ?
```



for b
- start timer
- go to neighbor of b