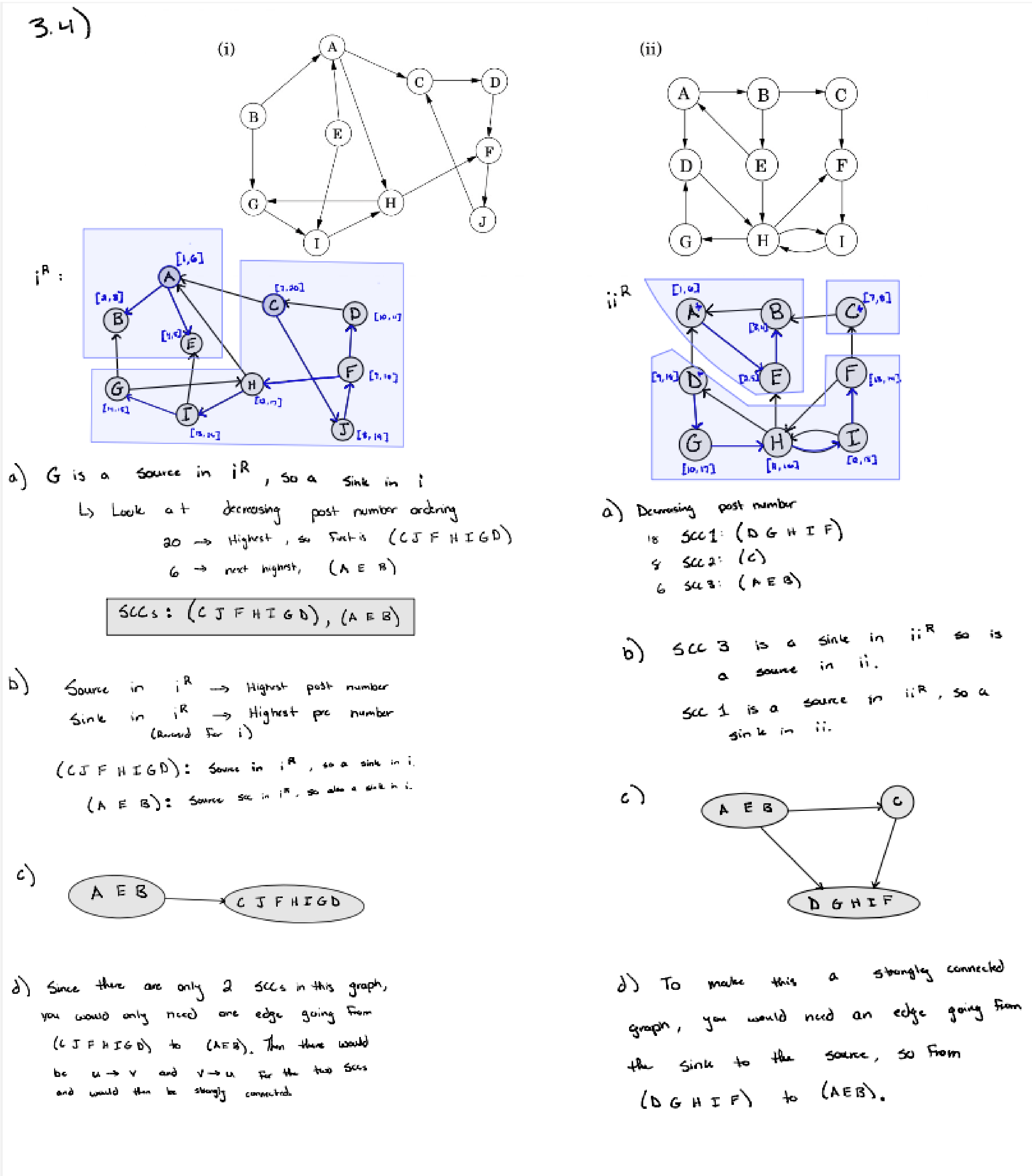
Intro to Algorithms: Homework #6

Due on March 19, 2021

Prof. Zaki

Jared Gridley



3.4)

(i)

i^R :

a) G is a source in i^R, so a sink in i

↳ Look at decreasing post number ordering

20 → Highest, so first is (C J F H I G D)

6 → next highest, (A E B)

SCCs: (C J F H I G D), (A E B)

b) Source in i^R → Highest post number

Sink in i^R → Highest pre number

(Reversed for i)

(C J F H I G D): Source in i^R, so a sink in i.

(A E B): Source scc in i^R, so also a sink in i.

c)

d) Since there are only 2 SCCs in this graph, you would only need one edge going from (C J F H I G D) to (AEB). Then there would be u → v and v → u for the two SCCs and would then be strongly connected.

(ii)

ii^R

a) Decreasing post number

18 SCC 1: (D G H I F)

9 SCC 2: (C)

6 SCC 3: (A E B)

b) SCC 3 is a sink in ii^R so is a source in ii.

SCC 1 is a source in ii^R, so a sink in ii.

c)

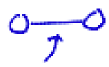
d) To make this a strongly connected graph, you would need an edge going from the sink to the source, so from (D G H I F) to (AEB).

Figure 1: Page 1

3.6) $d(u)$ is num of neighbors of vertex u.

a) Show in undirected graph, $\sum_{u \in V} d(u) = 2|E|$

The left side is saying that we sum up all the neighbors in each node in the graph. This is the Handshaking Theorum, where we know that each edge must start and end at a different vertex. So for each edge that we count, there will 2x as many neighbors.



For each edge, there are 2 vertices since it will be counted from each vertic

Cannot have this:

Also similar to combining a directed graph and its inverse, both have degree $|E|$, so when combining them you get $2|E|$.

b) If $\sum_{u \in V} d(u) = 2|E|$ is true (which we just showed), then if we had an odd number of vertices, our degree could not be odd because it would violate the rule in part (a), that is that any number x 2 must be even.

If we had an odd total degree, we would have a half an edge, which is not possible.

Cannot do:

cannot have half an edge.

c) No, in a directed graph, it follows the Directed Handshaking Theorum which states that $\sum_{i=1}^{n} \text{in-deg}(v_i) = \sum_{i=1}^{n} \text{out-deg}(v_i) = |E|$. So with an odd indegree it is possible to have an odd total degree

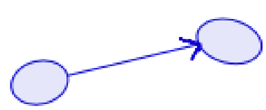


Figure 2: Page 2

3.9)

[[..........],
[..........],
[..........], ...
]

Adjacency List Format:

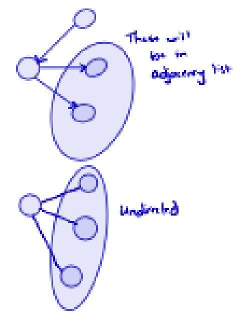
- Like a linked list
- Each vertex list contains vertices that have out edge with

twodegree (u):

```
degree = [ ]
two-degree = [0, 0, 0, ..., 0]

for i in range(u):
    c = count the elements in Vi's adjacency list
    degree[i] = c

For i in range(u):
    For j in range(vi):
        two-degree[i] += degree[j]

return two-degree
```

Algorithm:

For each node in each adjacency list, we go through and assign a degree value, and for each list add it to the two degree array.



3.11) Linear algorithm to determine if graph (undirected) e has a cycle.

DFS → with markers to see where we have already been → Start at vertex e.

If you find a cycle then return true, else false.

↳

Algorithm (G):

1) Do pre / post order numbering ← $O(V+E)$

2) During pre / post numbering

- Make a list, when a new node is visited, add it to the list, remove it when you return.

- If encounters a backedge, return True if vertex e is between current vertex and backedge node in list

[a, c, d, e, f, m]

keep looking, e isn't in there.

return True b/c e is in there

↳ we will need to traverse the list, which at worse case is the number of nodes.

So total time becomes $O(2V+E)$

↳ $O(V+E)$ → "Linear"

Figure 3: Page 3