CS: 3330 Exam 2 Solution

1(a) Pseudo code

```
// Compute degrees and set H[v] ← 1 if degree[v] ≥ 10
for each v ∈ V do
    count ← 0
    for each neighbor w of v do
        count ← count + 1
    degree[v] ← count
    if degree[v] ≥ 10 then
        H[v] ← 1
```

O(m+n) — Initialization

```
// Neighbors of vertices with degree ≥ 10 set their
// H[·] to 1
for each v ∈ V do
    if H[v] = 1 then
        for each neighbor w of v do
            H[w] ← 1
```

O(m+n) — Block 1

```
// Vertices 2 hops away from vertices with degree
// ≥ 10 set their H[·] to 1
for each v ∈ V do
    if H[v] = 1 then
        for each neighbor w of v do
            H[w] ← 1.
```

O(m+n) — Block 2

1(b) Note that the code in Block 1 and Block 2 are identical. Block 1 spreads $H[\cdot]=1$ to neighbors of vertices with degree ≥ 10. Block 2 spreads $H[\cdot]=1$ to vertices 2 hops aways from vertices with degree ≥ 10.

To spread this to vertices 20 hops away, we just need to enclose the block of code in Block 1 in a loop that executes 20 times.

(Since each blocks runs in $O(m+n)$ time, 20 executions of this block also runs in $O(m+n)$ time.)

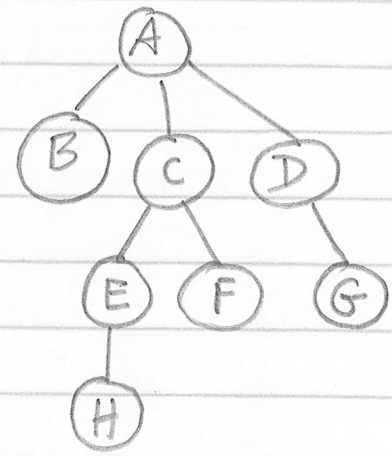
2(a) Delete $\{u,v\}$ from G and run the <u>explore</u> function with source u on the resulting graph. If v is visited by the explore function, then $\{u,v\}$ is in a cycle in G.

(<u>Explanation</u>: If there is a path from u to v in the graph without edge $\{u,v\}$, then adding $\{u,v\}$ to that path gives a cycle containing $\{u,v\}$.)

2(b)

Front ← A ← Back

B | C | D
B | D | E | F
B | E | F | G
B | F | G | H
B | G | H
B | H
B
ϕ



BFS Tree

3(a)

for all $u \in V$
  dist[u] ← ∞
  Prev[u] ← NIL
dist[s] ← c(s)

H ← makeQueue(V)
(with dist-values as keys

<u>while</u> H is non-empty <u>do</u>
  u ← deleteMin(H)

<u>for</u> all edges $\{u,v\} \in E$ <u>do</u>
  <u>if</u> dist[v] >
    dist[u] + $\ell(u,v)$ + c(v) <u>then</u>
    dist[v] ← dist[u] + $\ell(u,v)$ + c(v)
    prev(v) ← u
    decreaseKey(H, v)

3(b) (i)

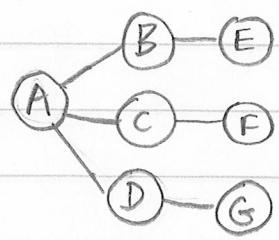
	D	A	B	C
Initially	0	∞	∞	∞
After Iter 1	0	1	2	4
After Iter 2	0	0	1	4
After Iter 3	0	−1	1	4

distances

(ii) An alternate edge ordering that allows Bellman-Ford to complete in 1 iteration:

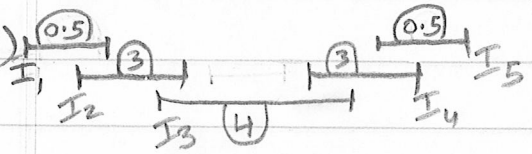
(D, C), (C, B), (B, A), (D, A), (D, B)

4(a)



A dominates 4 vertices, including itself. This is strictly more than the number of vertices dominated by any other vertex. So greedy picks A. After that greedy is forced to pick 3 more vertices, one to dominate E, one to dominate F, and one to dominate G. So the greedy dominating set has size 4. Optimal solution = {B, C, D}.

4 (b)



Intervals	$\omega(I)/\deg(I)$
I_1, I_5	$0.5/1 = 0.5$
I_2, I_4	$3/2 = 1.5$
I_3	$4/2 = 2$

So greedy picks I_3 and then I_1 & I_5. Total weight = 5
OPT = $\{I_2, I_4\}$. Weight of OPT = 6.