From the questions below, do exactly one of 3(a) and 3(b), as they are essentially equivalent formulations of the same problem.

1. (10pts) Show that the following language is $\text{NL}$-complete:

$$\{[G] : G \text{ is a strongly-connected digraph}\}.$$  

Note: A digraph $G$ is strongly-connected if for every ordered pair $u, v \in V(G)$ there is a directed path from $u$ to $v$.

2. (10pts) Define $\text{polyL}$ to be

$$\text{polyL} = \bigcup_{c \geq 1} \text{SPACE}[(\log n)^c].$$

Steve’s Class $\text{SC}$ (named in honor of Steve Cook) is defined to be the set of languages that can be decided by deterministic machines that run in polynomial time and $(\log n)^c$ space for some $c \geq 1$.

(a) (5pts) It is an open problem whether $\text{Reach}$ is in $\text{SC}$. Why does Savitch’s Theorem not resolve this question?

(b) (5pts) Is $\text{SC}$ the same as $\text{polyL} \cap \text{P}$?

3(a). (15pts) A linked list can be described as a digraph on vertices $\{1, \ldots, n\}$ with maximum out-degree one specified as an adjacency list. That is, every vertex $u$ has an out-going neighbor $p(u) \in \{0, 1, \ldots, n\}$, where $p(u) = 0$ when there is no edge leaving $u$.

Describe a log-space algorithm that takes an integer $n$ and a list of $n$ numbers representing $p(i)$ for all $i \in \{1, \ldots, n\}$, outputs 1 if the sequence $(a_i)_{i=1}^\infty$ defined as

$$a_1 = 1, \quad a_{i+1} = p(a_i)(i \geq 1, a_i \neq 0)$$

terminates (when $a_i = 0$ for some $i$) or continues forever ($a_i \neq 0$ for all $i$).

3(b). (15pts) Consider the following C structure.

```c
struct node {
    int value;
    struct node *next;
};
```

The `node` structure can store a linked list by following the `next` pointers to the next item in the list. Write a C function `bool hasCycle(struct node *list)` that returns `true` if and only if the linked list starting at `list` contains a cycle, but use a constant number of variables (i.e. do not use arrays or recursion). Prove that your function works correctly.

4. (Required for 824 students only; 15pts) Prove the **Space Hierarchy Theorem**: For $f, g : \mathbb{N} \to \mathbb{N}$ where $f$ and $g$ are space-constructible and $f(n) = o(g(n))$, then

$$\text{SPACE}[f(n)] \subsetneq \text{SPACE}[g(n)].$$

You should use the following fact (but you do not need to prove it): There exists a universal Turing machine $U$ that on input $\langle \alpha, x \rangle$, $U$ will simulate $M_\alpha$ acting on $x$. Moreover, if $M_\alpha$ uses $t$ work cells while computing $M(x)$, then $U(\alpha, x)$ uses $Ct$ work cells, where $C$ is a constant depending only on $M_\alpha$.  

1