Discussion Worksheet 7

1 Reference Counting

Exercise 1 Consider the following Java program performing some linked list operations:

```java
class Node {
    int value;
    Node next;
    void init(int value, Node next) {
        this.value = value;
        this.next = next;
    }
}
Node foo() {
    Node head = new Node(1);
    head.next = new Node(2);
    Node mid = head.next;
    mid.next = new Node(3);
    return head;
}
Node x = foo();
// line for part 1
_____    // line for part 2
x = null;
```

1.1 We want to perform reference counting garbage collection on the code. What is the reference count for each node object (designated by its value) at the line with the comment "line for part 1"?

Node(1): _____

Node(2): _____

Node(3): _____

1.2 Fill in the blank line with code that would make reference counting fail to reclaim all memory when the program finishes. Do not add new variables or create new objects. Give the reference counts after this change when the program finishes, after GC is performed. What weakness of reference counting does this expose?

2 Mark & Sweep

Exercise 2 Garbage collect the following heap using Mark & Sweep garbage collection. Clearly indicate which cells will be marked, and construct the free list resulting from the collection.
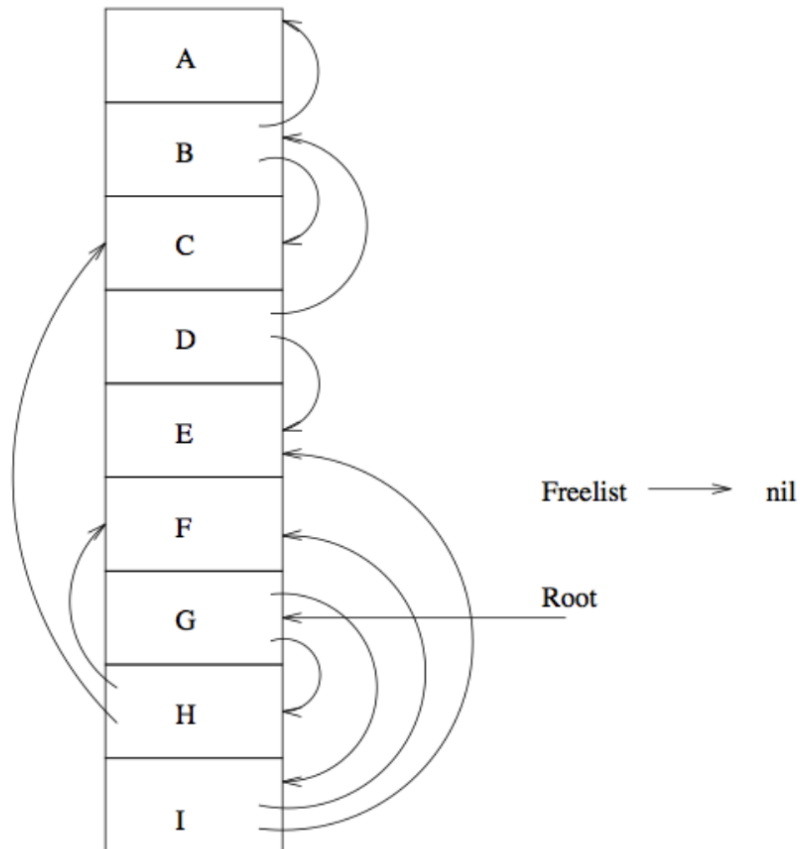
Marked: _____

Freelist: _____

3 Stop & Copy

Exercise 3 Show the result of running the Stop and Copy garbage collection algorithm from class on the heap below. Fill in the objects and pointers in the new heap. You need not represent the forwarding pointers from the old to the new space.

Some important points:

- The roots are processed in the order: acc, Frame 1

- When an object has several pointers to other objects, process the pointers in alphabetical order of destination.

OLD HEAP   NEW HEAP