

This homework is GROUP work. Total points is 16.

- (4 points) Write a C program that creates the situation on slide 8. Assume you have the following node C struct. Although the diagram shows only one local variable i, you can choose to use other local variables other than i. (Hint: You can do a series of `->next` to avoid using other variables.)

```
struct node { int n; struct node *next; };
```

```
int main (void)
{
    // i → A
    struct node* A = create_node();
    struct node* i = create_node();
    link_node(i,A);

    // i → A → B
    struct node* B = create_node();
    link_node(A,B);

    // i → A → B → D
    struct node* D = create_node();
    link_node(B,D);

    // i → A → B → D → C
    struct node* C = create_node();
    link_node(D,C);

    // i → A → B → D → C → B → D → C → B → ...
    link_node(C,B);
}

struct node* create_node(void)
{
    struct node* new_node = (struct node*) malloc(sizeof(struct node));
    new_node → n = 0;
    new_node → next = NULL;
    return new_node;
}

void link_node(struct node* head, struct node* tail)
{
    head → next = tail;
    tail → n = tail → n + 1;
}
```

- (4 points) Assume a machine only has 8 memory cells. Each memory cell can contain either a pointer `px` when `x` is the destination the pointer points to or a number `nx` where `x` is the actual number. The current memory content is shown below:

1	2	3	4	5	6	7	8
p2	n5	n2	n1	p8	n6	p3	p2

Which memory cells will be marked as unreachable in the mark stage? Demonstrate how you find them by drawing a diagram.

First we see what each memory cell references.

1:p2  
2:n5  
3:n2  
4:n1  
5:p8  
6:n6  
7:p3  
8:p2

Then we mark all references to cells

\* 1:p2 → 2:n5  
4:n1  
\* 5:p8 → 8:p2 → 2:n5  
6:n6  
\* 7:p3 → 3:n2

We then keep all marked cells, and remove all unmarked cells.

Keep:  
1:p2  
2:n5  
5:p8  
8:p2  
7:p3  
3:n2

Remove:  
4:n1  
6:n6

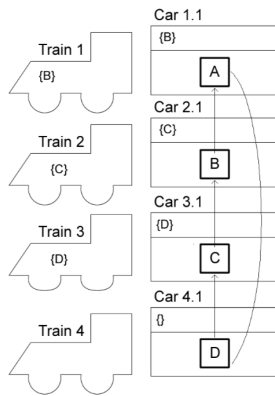
- (2 points) What is the advantage of generational garbage collection over mark and sweep?

Generational garbage collection is better than mark and sweep because generational garbage collection does not require as much time and does not fragment the data compared to mark and sweep because it defragments the heap as it runs and it only checks the younger parts of the heap

- (2 points) What is the disadvantage of generational garbage collection comparing to the train algorithm?

Generational garbage collection is not as great compared to the train algorithm because the generational garbage collection is not as good when it comes to retrieving older items from the heap compared to younger ones.

5. (4 points) Demonstrate how the following objects are collected (Figure 17 in the paper):



```

Step #1
Car 1[//Freed]
Car 2[A B]
^
Car 3[C]
^
Car 4[D]

Step #2
Car 1[//Freed]
Car 2[//Freed]
^
Car 3[A B C]
^
Car 4[D]

Step #3
Car 1[//Freed]
Car 2[//Freed]
Car 3[//Freed]
^
Car 4[A B C D]

Step #4
Car 1[//Freed]
Car 2[//Freed]
Car 3[//Freed]
Car 4[//Freed]

```