

# COP 2535: Data Structures, Exam 01

Single Linked Lists, Double Linked Lists, Circular Linked Lists, LRU Algorithm

## Instructions

This is an open book, open note test. All questions come from the readings in *Mastering Algorithms with C*. Otherwise, you may not give or receive aid. You may not use the internet.

Answer the following questions in writing. These are short answer questions. Upload your answers in Canvas as plain text. Please include your name in the text.

1. What are the advantages of linked lists over arrays?
2. What elements do a Singly Linked List contain? What are the contents to each element?
3. How do you traverse a Singly Linked List ? Can you traverse it in one direction or two?
4. What is the result if you drop an inner element from a Singly Linked List ?
5. Examine the following code. Explain in detail what lines 16 through 19 do.

```
1  int list_ins_next(List *list, ListElmt *element, const void *data)
2  {
3      ListElmt      *new_element;
4      if ((new_element = (ListElmt *)malloc(sizeof(ListElmt))) == NULL)
5          return -1;
6      new_element->data = (void *)data;
7      if (element == NULL)
8      {
9          if (list_size(list) == 0)
10             list->tail = new_element;
11             new_element->next = list->head;
12             list->head = new_element;
13         }
14     else
15     {
16         if (element->next == NULL)
17             list->tail = new_element;
18             new_element->next = element->next;
19             element->next = new_element;
20         }
21     list->size++;
22     return 0;
23 }
```

6. What are the differences between a Singly Linked List and a Doubly Linked List ?
7. What are the elements of a Doubly Linked List ? What are the contents of each element?

8. Examine the following code. Explain in detail what lines 18 through 24 do.

```
1  int dlist_ins_next(DList *list, DListElmt *element, const void *data)
2  {
3      DListElmt      *new_element;
4      if (element == NULL && dlist_size(list) != 0)
5          return -1;
6      if ((new_element = (DListElmt *)malloc(sizeof(DListElmt))) == NULL)
7          return -1;
8      new_element->data = (void *)data;
9      if (dlist_size(list) == 0)
10     {
11         list->head = new_element;
12         list->head->prev = NULL;
13         list->head->next = NULL;
14         list->tail = new_element;
15     }
16     else
17     {
18         new_element->next = element->next;
19         new_element->prev = element;
20         if (element->next == NULL)
21             list->tail = new_element;
22         else
23             element->next->prev = new_element;
24         element->next = new_element;
25     }
26     list->size++;
27     return 0;
28 }
```

9. What are the differences between a Singly Linked List and a Circular Linked List ? What are the differences between a Doubly Linked List and a Circular Linked List ?
10. Write the pseudocode for the Least Recently Used algorithm as it is implemented by a Circular Linked List . The book calls this the *clock algorithm*. If you wish, you may write a *detailed* explanation of the Least Recently Used algorithm rather than pseudocode.