These notes were prepared by Carl.

## Stacks (LIFO Queues):

Applications:

Implementation:

Linked List: Drawing:

|  |  |
| --- | --- |
| typedef struct node \* nodePT;  struct node {  int item; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  int \* next; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  }; |  |

Array: Drawing:

|  |  |
| --- | --- |
| struct stack\_array {  int \* items; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  int top; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  int capacity; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  }; |  |

Operations:

void push(Stack \* s, int item):

TC: \_\_\_\_\_\_\_\_\_\_\_ SC: \_\_\_\_\_\_\_\_\_\_\_

int pop(Stack \* s):

TC: \_\_\_\_\_\_\_\_\_\_\_ SC: \_\_\_\_\_\_\_\_\_\_\_

Shorthand:

x = push(x)

\* = pop()

Ex:

a.) Implementation method: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operations List: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b.) Implementation method: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operations List: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

If letters can only be pushed onto the stack once, is it possible to create desired output?

a.) Available Letters for Input: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Desired Output: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operations Required: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b.) Available Letters for Input: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Desired Output: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operations Required: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## FIFO Queues:

Applications:

Implementation:

Linked List: Drawing:

|  |  |
| --- | --- |
| typedef struct node \* nodePT;  struct queue\_list {  nodePT first; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  nodePT last; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  int size; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  }; |  |

Array: Drawing:

|  |  |
| --- | --- |
| struct queue\_array {  int capacity; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  int size; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  int first\_index; \_\_\_\_\_\_\_\_\_\_\_\_\_  int last\_index; \_\_\_\_\_\_\_\_\_\_\_\_\_\_  int \* items; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  }; |  |

Operations:

bool put(Queue \* Q, int item):

TC: \_\_\_\_\_\_\_\_\_\_\_ SC: \_\_\_\_\_\_\_\_\_\_\_

bool get(Queue \* Q, int\* return):

TC: \_\_\_\_\_\_\_\_\_\_\_ SC: \_\_\_\_\_\_\_\_\_\_\_

Shorthand:

x = put(x)

\* = get()

Ex:

a.) Implementation method: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Queue Size: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operations List: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b.) Implementation method: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Queue Size: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operations List: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c.) Implementation method: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Queue Size: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operations List: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d.) Implementation method: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Queue Size: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operations List: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Possible Actions When Queue is Full:

* Rejecting the Insertion:

Why:

* Increasing Size of Queue

Why:

Reallocation for Larger Queue:

Considerations:

Time Complexity:

Space Complexity:

Linked List:

Types:

Accessing & Searching Elements in Linked List:

Iterating through Linked List:

NULL case:

Adding/Removing Nodes:

Destroying Linked List:

## 

## 