Rightward Diff

Do Not Use

Linear Any
index = index % size of array
++index
++index
Linear Array

\[ \begin{array}{c}
index = \frac{6 \times 6}{\phi} \\
\phi = 15 \\
++index
\end{array} \]

Circular Array
Data Structures

Manner of Operation:

- Init size of array:
  int size = size_of_array;

- Init number of items:
  int number_of_items;

- Init front:
  int front;

- Init rear:
  int rear;

- Data array:
  data * array;

- Private:
  Circular Array...
dynamically allocated

- Rear:
  - rear = rear % size_of_array + 1;
- Front:
  - front % size_of_array + 1;

They wrap around back to the beginning

- When the "rear" or "front" indices progress to the end,
Data Structures
Data Structures

Linked List:

- Tail = NULL;
- Tail -> next = NULL;
- Tail -> member = data;
- Tail -> next = new node;
- Tail = tail -> next;

Array:

Array = [data];

Circular array:

How does the linked list compare with the array?

- Yes! Otherwise traversals would have been required
- No! Did we need a tail pointer?
One Item

Empty List

CELL

null
Add an item to a non-empty list:

```
20
{ 0  

node = new node
pr = pr -> next
pr = pr
pr = pr
```

Add an item to a non-empty list:

```
null
```

Add an item to a non-empty list:

```
```
```c
5 + 5 \approx 10

```
Doubly Linked Lists