Lec-24

12/07/25 12:55 PM

Data Str

① Stack
② Queue

Stack → class → non primitive → heap

Stack → LIFO → Last in first out → Application

idx++ arr[idx] = item

arr[idx+1] = item

40
30
20
10

LIFO

add → push
delete → pop
get → peek
size → size
isEmpty()

push(10)
push(20)
push(30)
push(40)

idx = -1

0

10

0 1 2 3 4

idx++
size

idx = -1 0 1 2 3 4

10 20 30 40 50

0 1 2 3 4

pop() {
x = arr[idx]
idx--
return x
}

peek() {
return arr[idx]
}

push(10)
push(20)
push(30)
pop()
push(40)
push(50)
pop()
peek()

Size = 0 1 2 3 4
front = 0

Queue

10 20 30 40

Enq(10)
Enq(20)
Enq(30)
Enq(40)
deq()
deq()
Enq(50)

front

rear

20 30 40 50

FIFO
LILO

get
remove

Add

Add → Enqueue → arr[size] = item, size++
remove → dequeue
set → set front → x = arr[front], front++, size--

```
// O(1)
    public void Enqueue(int item) {
        arr[size] = item;
        size++;
    }

    // O(1)
    public int Dequeue() {
        int x = arr[front];
        front++;
        size--;
        return x;
    }
```

60 30 40 50

0 1 2 3 4

9 4 5 6 7 8

30 40 50 60

front + size = 2 + 2 = 4

= 2 + 3 = 5

Enq(10)
Enq(20)
Enq(30)
Enq(40)
deq()
deq()
Enq(50)
Enq(60)

Size = 0 1 2 3 4
front = 0 1 2

front

60 30 40 50

0 1 2 3 4

30 40 50 60

for(i=0; i<size; i++){
idx = (f+i) % arr

i=0 (2+0)%5 = 2
i=1 (2+1)%5 = 3
i=2 (2+2)%5 = 4
(2+3)%5 = 0

30
20
10

10

30 20 10 7

```
private static void Add_Last(Stack<Integer> st, int item) {
    // TODO Auto-generated method stub
    if (st.isEmpty()) {
        st.push(item);
        return;
    }
    int x = st.pop();
    Add_Last(st, item);
    st.push(x);
}
```

30
20
10
7

x = 10
(2x, 7)
St.push(x)

x = 20
(2x, 7)

x = 30
(2x, 7)

[2x, 7]

St = 2x
item = 7

stack

{ [( )] ( ) }