5560. Design Front Middle Back Queue

My Submissions (/contest/biweekly-contest-40/problems/design-front-middle-back-queue/submissions/) Back to Contest (/contest/biweekly-contest-40/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Design a queue that supports `push` and `pop` operations in the front, middle, and back.

Implement the `FrontMiddleBack` class:

- `FrontMiddleBack()` Initializes the queue.
- `void pushFront(int val)` Adds `val` to the **front** of the queue.
- `void pushMiddle(int val)` Adds `val` to the **middle** of the queue.
- `void pushBack(int val)` Adds `val` to the **back** of the queue.
- `int popFront()` Removes the **front** element of the queue and returns it. If the queue is empty, return `-1`.
- `int popMiddle()` Removes the **middle** element of the queue and returns it. If the queue is empty, return `-1`.
- `int popBack()` Removes the **back** element of the queue and returns it. If the queue is empty, return `-1`.

Notice that when there are **two** middle position choices, the operation is performed on the **frontmost** middle position choice. For example:

- Pushing `6` into the middle of `[1, 2, 3, 4, 5]` results in `[1, 2, 6, 3, 4, 5]`.
- Popping the middle from `[1, 2, 3, 4, 5, 6]` returns `3` and results in `[1, 2, 4, 5, 6]`.

Example 1:

```
Input:
["FrontMiddleBackQueue", "pushFront", "pushBack", "pushMiddle", "pushMiddle", "popFront", "popMiddle", "popMiddle", "popBack",
[[], [1], [2], [3], [4], [], [], [], [], []]
Output:
[null, null, null, null, null, 1, 3, 4, 2, -1]

Explanation:
FrontMiddleBackQueue q = new FrontMiddleBackQueue();
q.pushFront(1);   // [1]
q.pushBack(2);    // [1, 2]
q.pushMiddle(3);  // [1, 3, 2]
q.pushMiddle(4);  // [1, 4, 3, 2]
q.popFront();     // return 1 -> [4, 3, 2]
q.popMiddle();    // return 3 -> [4, 2]
q.popMiddle();    // return 4 -> [2]
q.popBack();      // return 2 -> []
q.popFront();     // return -1 -> [] (The queue is empty)
```

Constraints:

- `1 <= val <= 10^9`
- At most `1000` calls will be made to `pushFront`, `pushMiddle`, `pushBack`, `popFront`, `popMiddle`, and `popBack`.

JavaScript

```
function FrontMiddleBackQueue() {
    this.q = [];
};

FrontMiddleBackQueue.prototype.pushFront = function (val) {
    this.q.unshift(val);
};

FrontMiddleBackQueue.prototype.pushMiddle = function (val) {
    let n = this.q.length;
    let idx = n >> 1;
    let tmp = this.q.slice(0, idx).concat([val]).concat(this.q.slice(idx));
    this.q = tmp;
};

FrontMiddleBackQueue.prototype.pushBack = function (val) {
    this.q.push(val);
};
```

```
19
20 FrontMiddleBackQueue.prototype.popFront = function () {
21     let n = this.q.length;
22     if (n == 0) return -1;
23     let res = this.q[0];
24     this.q.shift();
25     return res;
26 };
27
28 FrontMiddleBackQueue.prototype.popMiddle = function () {
29     let n = this.q.length;
30     if (n == 0) return -1;
31     let idx;
32     if (n % 2 == 1) {
33         idx = n >> 1;
34     } else {
35         idx = (n >> 1) - 1;
36     }
37     let res = this.q[idx];
38     let tmp = this.q.slice(0, idx).concat(this.q.slice(idx + 1));
39     this.q = tmp;
40     return res;
41 };
42
43 FrontMiddleBackQueue.prototype.popBack = function () {
44     let n = this.q.length;
45     if (n == 0) return -1;
46     let res = this.q[n - 1];
47     this.q.pop();
48     return res;
49 };
50
51
52 /**
53  * Your FrontMiddleBackQueue object will be instantiated and called as such:
54  * var obj = new FrontMiddleBackQueue()
55  * obj.pushFront(val)
56  * obj.pushMiddle(val)
57  * obj.pushBack(val)
58  * var param_4 = obj.popFront()
59  * var param_5 = obj.popMiddle()
60  * var param_6 = obj.popBack()
61  */
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/425009004/) | More Details > (/submissions/detail/425009004/)

Share your acceptance!

Copyright © 2020 LeetCode | Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)