Dynamic Array

29 more points to get your next star!

Rank: 2141259 | Points: 71/100

Problem Solving

Your Dynamic Array submission got 15.00 points. Share Post

You are now 29 points away from the 2nd star for your problem solving badge.

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

- Declare a 2-dimensional array, arr, with n empty arrays, all zero-indexed.
- Declare an integer, $lastAnswer$, and initialize it to 0.

You need to process two types of queries:

1. Query: $1\ x\ y$
   - Compute $idx = (x \oplus lastAnswer)$.
   - Append the integer y to $arr[idx]$.
2. Query: $2\ x\ y$
   - Compute $idx = (x \oplus lastAnswer)$.
   - Set $lastAnswer = arr[idx][y\%size(arr[idx])]$.
   - Store the new value of $lastAnswer$ in an answers array.

Notes:

- $\oplus$ is the bitwise XOR operation, which corresponds to the ^ operator in most languages. Learn more about it on Wikipedia.
- $\%$ is the modulo operator.
- Finally, $size(arr[idx])$ is the number of elements in $arr[idx]$.

Function Description

Complete the $dynamicArray$ function with the following parameters:

- $int\ n$: the number of empty arrays to initialize in arr
- $int\ queries[q][3]$: 2-D array of integers

Returns

- $int[]$: the results of each type 2 query in the order they are presented

Input Format

The first line contains two space-separated integers, n, the size of arr to create, and q, the number of queries, respectively.

Each of the q subsequent lines contains a query string, $queries[i]$.

Constraints

- $1 \leq n, q \leq 10^5$
- $0 \leq x, y \leq 10^9$
- It is guaranteed that query type 2 will never query an empty array or index.

Sample Input

```
STDIN   Function
-----   --------
2 5     size of arr[] n = 2, size of queries[] q = 5
1 0 5   queries = [[1,0,5],[1,1,7],[1,0,3],[2,1,0],[2,1,1]]
1 1 7
1 0 3
2 1 0
2 1 1
```

Sample Output

```
7
3
```

Explanation

Initial Values:

$n = 2$

$lastAnswer = 0$

$arr[0]$ = [ ]

$arr[1]$ = [ ]

Query 0: Append 5 to $arr[(\ (0 \oplus 0)\ \%\ 2\ )] = arr[0]$.

$lastAnswer = 0$

$arr[0]$ = [5]

$arr[1]$ = [ ]

Query 1: Append 7 to $arr[(\ (1 \oplus 0)\ \%\ 2\ )] = arr[1]$.

$arr[0]$ = [5]

$arr[1]$ = [7]

Query 2: Append 3 to $arr[(\ (0 \oplus 0)\ \%\ 2\ )] = arr[0]$.

$lastAnswer = 0$

$arr[0]$ = [5, 3]

$arr[1]$ = [7]

Query 3: Assign the value at index 0 of $arr[(\ (1 \oplus 0)\ \%\ 2\ )] = arr[1]$ to $lastAnswer$. Store $lastAnswer$ in your answer array. $lastAnswer = 7$

$arr[0]$ = [5, 3]

$arr[1]$ = [7]

Query 4: Assign the value at index 1 of $arr[(\ (1 \oplus 7)\ \%\ 2\ )] = arr[0]$ to $lastAnswer$. Store $lastAnswer$ in your answer array. $lastAnswer = 3$

$arr[0]$ = [5, 3]

$arr[1]$ = [7]

Return your answer array [7, 3]. The code stub prints its elements on separate lines.

Change Theme Language Python 3

```
17
18  def dynamicArray(n, queries):
19      # Write your code here
20      # Do not use arr = [[]] * n
21      arr = []
22      for _ in range(n):
23          arr.append([])
24      last_answer = 0
25      answers = []
```

```
    answers = []
    for q in queries:
        q_type, x, y = q
        if q_type == 1:
            idx = (x ^ last_answer) % n
            arr[idx].append(y)
        if q_type == 2:
            idx = (x ^ last_answer) % n
            last_answer = arr[idx][y%len(arr[idx])]
            answers.append(last_answer)

    return answers

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    first_multiple_input = input().rstrip().split()

    n = int(first_multiple_input[0])

    q = int(first_multiple_input[1])

    queries = []

    for _ in range(q):
        queries.append(list(map(int, input().rstrip().split())))
```

EMACS Line: 33 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

You have earned 15.00 points!

You are now 29 points away from the 2nd star for your problem solving badge.

59% 71/100

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0 | Test case 1 | Test case 2 | Test case 3 | Test case 4

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hackos.

Test case 5

Test case 6

Unlock