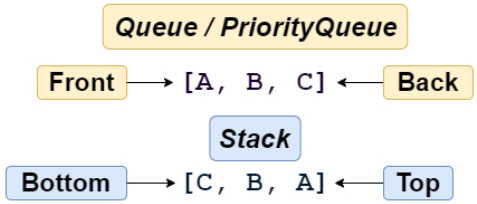
Data Structure Matching

Marcar esta página

Given a log of **add()** and **remove()** calls on an unknown data structure, determine what the data structure is. Options include: **Stack**, **Queue**, and **PriorityQueue**. If the result is not possible, select **None**. Please note that there may be more than one correct answer for each question. If that is case, select **all** possible answers.

When referring to a PriorityQueue, prioritize elements based on the natural ordering of the letters. i.e. 'A' has higher priority than 'B', 'C' has higher priority than 'D', etc.

When analyzing the resulting data structure state, consider the left side of the list to represent the front of a Queue and the right side of the list to represent the back of a Queue. Additionally, consider the left side of the list to represent the bottom of a Stack and the right side of the list to represent the top of a Stack. You are provided with a scratchpad at the bottom of this page that you may use.



Data Structure Matching

0 puntos posibles (no calificables)

Question 1

```
Initial state of the data structure: []
```

```
add(A);
add(G);
remove();
add(E);
add(D);
remove();
```

```
After: [A, E]
```

- [x] Stack
- [ ] Queue
- [ ] PriorityQueue
- [ ] None

Question 2

```
Initial state of the data structure: []
```

```
add(A);
add(D);
remove();
add(G);
add(I);
remove();
```

```
After: [G, I]
```

- [ ] Stack
- [x] Queue
- [x] PriorityQueue
- [ ] None

✔

Question 3

```
Initial state of the data structure: []
```

```
add(A);
add(D);
remove();
add(E);
add(J);
remove();
```

```
After: [D, E]
```

- [ ] Stack
- [ ] Queue
- [ ] PriorityQueue
- [x] None

✔

Question 4

```
Initial state of the data structure: []
```

```
add(B);
remove();
add(D);
remove();
add(B);
add(D);
```

```
After: [B, D]
```

- [x] Stack
- [x] Queue
- [x] PriorityQueue
- [ ] None

✔

Question 5

```
Initial state of the data structure: []
```

```
add(C);
remove();
add(A);
add(C);
add(F);
remove();
```

```
After: [C, F]
```

- [ ] Stack
- [x] Queue
- [x] PriorityQueue
- [ ] None

✔

Question 6

```
Initial state of the data structure: []
```

```
add(A);
remove();
add(C);
remove();
add(G);
add(D);
```

```
After: [G, D]
```

- [x] Stack
- [x] Queue
- [ ] PriorityQueue
- [ ] None

✔

Enviar

Mostrar respuesta

Las respuestas son mostradas en el problema

Scratchpad

Below is a textbox you can use as a scratchpad while you work through this problem; you should be able to click in the bottom-right and drag to expand it for your use. No text in this box will be used by the grading script, but you can use it to take notes and work through the exercise. Note that text entered in this box **will not be saved** if you leave and return to this page.

< Anterior Siguiente >