

Questão 1

Dada a estrutura de pilha, ilustre o estado da pilha após a execução sequencial das seguintes operações: (OBS: O estado B leva em conta as operações feitas no estado A)

- (a) push(10) -> push(70) -> push(25) -> push(-50);
 (b) pop() -> pop();
 (c) push(13) -> pop() -> pop() -> push(10).

Pilha: "LIFO"

Push: insert
 Pop: remove

a)

-50
25
70
10

b)

70
10

c)

10
10

Questão 2

Dada a estrutura de fila, ilustre o estado da fila após a execução sequencial das seguintes operações: (OBS: O estado B leva em conta as operações feitas no estado A)

- (a) enqueue(10) -> enqueue(70) -> enqueue(25) -> enqueue(-50);
 (b) dequeue() -> dequeue();
 (c) enqueue(13) -> dequeue() -> dequeue() -> enqueue(10).

Fila: "Fifo"

Enqueue: insert
 Dequeue: remove

a)

0	1	2	3
10	70	25	-50

b)

0	1
25	-50

ou

0	1	2	3
25	-50		

c)

0	1
13	10

ou

0	1	2	3
13	10		

Questão 3

Dada a estrutura de fila implementada com um array de tamanho 4, ilustre o estado do array após a execução sequencial das seguintes operações: (OBS: O estado B leva em conta as operações feitas no estado A)

- (a) enqueue(10) -> enqueue(70) -> enqueue(25) -> enqueue(-50);
 (b) dequeue() -> dequeue();
 (c) enqueue(13) -> dequeue() -> dequeue() -> enqueue(10).

Array circular

a)

Head			Tail
10	70	25	-50

b)

	Head	Tail	
	25	-50	

c)

Head	tail		
13	10		

Questão 4

Inversão de uma fila usando uma pilha

Dada uma fila L , escreva um algoritmo que gere uma nova fila R contendo os elementos de L na ordem reversa utilizando uma estrutura de pilha.

Exemplo:

Entrada: $L = [A, B, C, D]$

Saída esperada: $R = [D, C, B, A]$

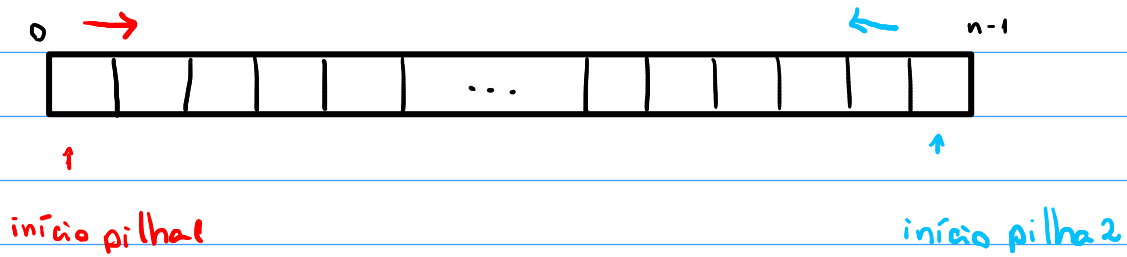
Pseudocódigo: Queue reverse (Queue F) {
 Stack S = initializeStack(F.size); $O(1)$ ou $O(n)$
 Queue R = initializeQueue(F.size); $O(1)$ ou $O(n)$
 while (F.size > 0) {
 int v = F.dequeue(); Arr normal: $O(n)$; Arr circ: $O(1)$
 S.push(v); $O(1)$
 }
 while (S.size > 0) {
 int v = S.pop(); $O(1)$
 R.enqueue(v); $O(1)$
 }
 return R;
}

$$\Rightarrow \sum_{i=0}^{n-1} (n+1) + \sum_{i=0}^{n-1} (2) = n^2 + 4n \Rightarrow O(n^2) \text{ array normal.}$$
$$\Rightarrow 2 \cdot \sum_{i=0}^{n-1} (2) = 4n \Rightarrow O(n) \text{ array circular.}$$

Questão 5

Explique como implementar duas pilhas em um único array de maneira eficiente.

A ideia seria controlar o meio do array, ou seja, a primeira pilha começa no início do array e a segunda pilha começa no final do array. Assim, enquanto a soma dos tamanhos das pilhas for \leq ao tamanho do array, podemos fazer operações, caso contrário, damos uma exceção.



Questão 6

Implemente a busca sequencial em uma lista circular.

Buscar um valor

Estrutura:

$\Rightarrow O(n)$

```
busca-seq(F, x) {
    if (F.Head ≤ F.Tail) {
        for (inti = F.Head; i ≤ F.Tail; i++) {  $O(n)$ 
            ... (normal)
        }
    } else {
        for (inti = 0; i ≤ F.Tail; i++) {  $O(n)$ 
            ... (normal)
        }
        for (inti = F.Head; i ≤ F.MaxSize; i++) {  $O(n)$ 
            ... (normal)
        }
    }
}
```

Questão 8

Mostre como implementar uma fila utilizando duas pilhas. Analise o tempo de execução das operações.

Armazeno a ordem de entrada em uma pilha e crio outra empilhando "ao contrário", ou seja, o topo da primeira (o que entrou agora na fila) é o topo da primeira pilha e a base da segunda pilha.

Exemplo: Considere a fila: 1 chegou 1º, 2 chegou 2º, 3 chegou 3º e 4 chegou 4º.

A primeira pilha é 4321 (topo é 4 - último a chegar) e a segunda é 1234 (topo 1 - primeiro a chegar.)

Questão 10

Agora, dado uma fila, implemente a operação `getMinValue`, que retorna o menor valor na fila:

(a) Implemente a função com complexidade $O(n)$

(b) Se soubéssemos que a fila só receberia valores no intervalo $[1, 10]$, seria possível implementar uma solução mais eficiente? Explique como otimizá-la.

a) Busca sequencial normal ou numa file circular.

b) Crie um array de inteiros que conta os valores 1,2,3,...,10 que têm na fila. Por exemplo: suponha que a fila é [9, 10, 5, 4, 8, 7, 3, 2, 1, 6, 8, 9, 10], assim o array auxiliar é [1, 1, 1, 1, 1, 1, 1, 2, 2, 2] que é a quantidade de cada número de 1 a 10 presentes na fila. Assim, para achar o menor valor, devemos percorrer o array auxiliar até que uma das entradas seja diferente de 0 e, como o array é fixo, isso é feito com complexidade $O(1)$.