u évora
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

EINF MA CID

Disciplina: Estruturas de Dados e Algoritmos I-2019/2020 Prova: mini-teste 1 (15-10-2019)

Esta prova tem a duração de **1 hora** e é **sem consulta**. Identifique TODAS as folhas de teste.

1. Considere o seguinte código Java:

```
public static boolean q1(long n){
        long c=n;
        StackArray<Integer> s= new StackArray<>();

        while (n>0){
            s.push((int)n%10);
            n=n/10;
        }

        while(!s.empty()){
            if(s.pop()!=c%10)
                return false;
            c=c/10;
        }
        return true;
    }
```

(a) Qual o resultado da execução de `q1(7)`?

(b) Qual o resultado da execução de `q1(213)`?

(c) Qual a complexidade do método `q1`?

2. Considere o seguinte código Java:

```
 public static void  q2(int n){
     StackArray<Long> q=new StackArray<Long>();
      q.push(0);
      q.push(1);
      for (int i=0;i<n;i++) {
          Long a=q.pop();
          Long b= q.pop()
          System.out.print(b+" ");
          q.push(a);
          q.push(a+b);
      }
}
```

(a) Qual o resultado da execução de `q2(9)`?

(b) Qual a complexidade do método `q2`?

3. Considere que no código de *QueueArray*, i.e a sua implementação de Queue (a interface para Queue), usando a técnica dos arrays circulares, a variável de instância q é o array usado, e as variáveis (também de instância) *ini* e *fim* representam respectivamente o início da queue e a próxima posição do array onde se farão as inserções.,implementou o método q3:

```
public void q3(){
        System.out.print("[");
        int i=ini;
        int n=0;
        while (i != fim){
            if (n++<size()-1)
                System.out.print(q[i]+";");
            else
                System.out.print(q[i]);
            i=(i+1)%q.length;
        }
        System.out.println("]");
    }
```

(a) Qual o resultado de x.q3(), se x for a queue $\{1, 10, 20\}$, sendo 1 o 1º elemento da fila?

(b) Se x for uma variável do tipo Queue, qual o resultado de x.q3()?

(c) Qual a complexidade do método?

4. Considere o seguinte código Java:

```
public static <E> Queue<E> q4(Queue<E> q){
        Stack<E> s=new StackArray<>(  );
        Queue<E> aux=new QueueArray<>( q.size());
        for(int i=0;i<q.size();i++)
            s.push(q.dequeue());
        for (int i=0;i<s.size();i++)
            aux.enqueue( s.pop() );
        return aux;
    }
     public static void main(String[] args){
        Queue<Integer> q1=new QueueArray<>(  );
        q1.enqueue( 2 );
        q1.enqueue( 1 );
        q1.enqueue( 1 );
        q1.enqueue( 2 );
        System.out.println("q1="+q1);
        Queue<Integer> u=Q3(q1);
        System.out.println("u="+u);
        System.out.println("q1="+q1);


    }
}
```

(a) Qual o 1º output gerado?

(b) Qual o 2º output gerado?

(c) Qual o 3º output gerado?

5. Suponha que quatro carruagens numeradas de 1 a 4 estão posicionadas à entrada duma estação ferroviária (ver figura) Suponha possíveis as seguintes acções:

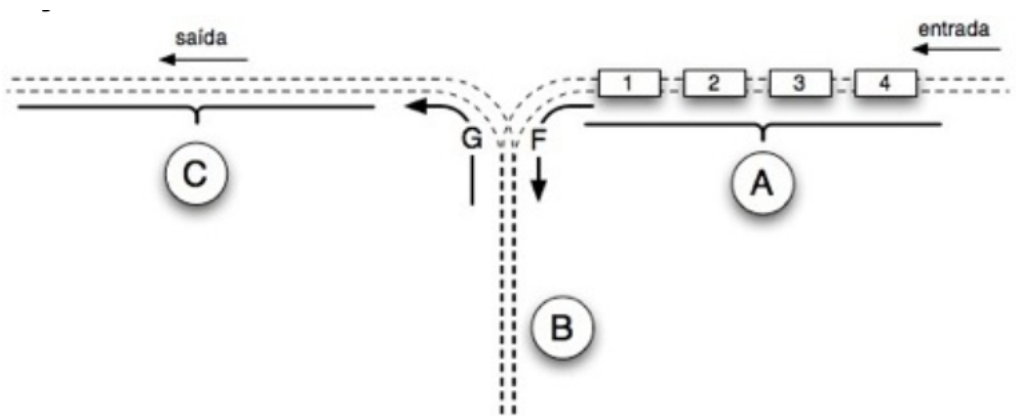

Figure 1: Estação Ferroviária

- *entrada(carruagem)*, que permite a entrada duma carruagem na estação(pela zona A)
- *saida()*, que permite que uma carruagem saia da estação (pela zona C)
- *F()* que permite que uma carruagem passe da zona A para a zona B
- *G()* que permite que uma carruagem passe da zona B para a zona C

(a) A sequência de acções: entrada(1), entrada(2),entrada(3), entrada(4), F(), F(), F(), F(), G(), G(), G(), G(), saida(), saida(), saida(), saida(), permitirá retirar da estação as carruagens pela ordem 4,3,2,1. Assumido que estão posicionadas na estação as carruagens na zona de entrada, na sequência (3,2,1), que sequência de operações deverá realizar para retirar da estação a carruagem 3?

(b) Apresente um programa em Java que modele o comportamento da estação, exibindo a codificação das operações acima descritas. Não pode usar arrays.