u évora
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

EINF MA CID

Disciplina: Estruturas de Dados e Algoritmos I-2018/2019 Prova: Teste 1 (5-11-2018)

Esta prova tem a duração de **2 horas** e é **sem consulta**. Identifique TODAS as folhas de teste.

1. Um palíndromo é um texto que é lido de igual modo, tanto da esquerda para a direita como da direita para a esquerda. Nos palíndromos ignoram-se os espaços, sinais de pontuação e os caracteres acentuados são substituídos pelo correspondente caracter sem a acentuação. Exemplo dum palíndromo em português: "Socorram-me, subi no ônibus em Marrocos". Implemente o método `public static boolean palindrome (String txt)` que retorna true, se o texto `txt` for um palíndromo. Não necessita de verificar(eliminar) espaços, pontuação, ou caracteres especiais; assuma que `txt` já está tratada. Está proibido de usar arrays e outras estruturas de dados com excepção das Stacks. Assuma as *Strings* iteráveis, para ir obtendo os caracteres da String.

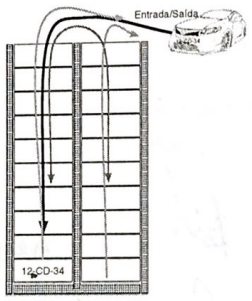

Figure 1: Parque de estacionamento

2. Um parque de estacionamento para automóveis, contem dois corredores de estacionamento, lado a lado, cada um com capacidade para 100 carros(ver figura 1). Em cada corredor, só os carros mais perto da saída podem sair. Os carros mais atrás estão bloqueados. Se um destes carros quiser sair pode ser usado o corredor lateral para desbloquear a fila de carros que estejam a tapar a saída. O próximo carro a chegar ocupa a posição do carro que saiu. As regras(entrar/sair) no corredor lateral, são exactamente as mesmas.

   (a) Implemente o parque de estacionamento; Somente as operações de entrada de carros no parque e saída de carros do parque.
   (b) Qual a capacidade do parque? 101 para poder trocar entre as filas
   (c) Qual o pior caso, para ambas as operações? tem que haver trocas entre
   (d) Qual o melhor caso, para ambas as operações? irão encher as 2 filas
   (e) Qual a complexidade das operações no pior dos casos? imediatamente
   # trocas

3. Apresente uma implementação para *Queues*, usando a técnica das listas ligadas. Defina na classes das Queues uma inner class como os nós que servem de suporte às suas listas. Indique, justificando, para cada uma das operações a complexidade temporal das mesmas.

4. Uma forma possível para implementar *Queues* usando listas ligadas é usar listas ligadas(simples) circulares. Nas listas ligadas circulares, a referência next do último nó da lista aponta para o primeiro nó na lista. Assuma que a lista não contem "dummy nodes", e que referencia um único nó na lista. Será possível implementar as operações básicas das *queues* em tempo constante se?

   (a) O nó referenciado for o primeiro nó da lista

   (b) O nó referenciado for o último nó da lista

5. Considere o seguinte código Java:

```
public static void  F(int n){
        QueueArray<Long> q=new QueueArray<Long>();
        q.enqueue(0);
        q.enqueue(1);
        for (int i=0;i<n;i++) {
            Long t = q.dequeue();
            System.out.print(t+" ");
            q.enqueue(t+q.front());

        }
}
```

   (a) Qual o resultado da execução de `F(8)`?

   (b) Qual a complexidade do método F