

Prova AEDs 2

Lucas Santiago de Oliveira

Março 2020

1 Questão 1

O pior caso: $3n + (\frac{n}{2} - 1) + (n - 7) + \log_2(n - 2) + (3n) + (2n^2 - 1)$.
Em notação: $\mathcal{O}(n^2)$.

O melhor caso: $n + (\frac{n}{2} - 1) + (n - 7) + \log_2(n - 2) + n + (2n^2 - 1)$.
Em notação: $\mathcal{O}(n^2)$.

2 Questão 2

```
public void inserirDir(int x){  
  
    if(esq + dir >= array.length())  
        MyIO.println("Erro ao inserir: deque cheio!");  
  
    else{  
        int posDir = array.length() - dir;  
  
        for(int i = dir; i < array.length(); i++)  
            array[dir - 1] = array[dir];  
  
        array[array.length() - 1] = x;  
        dir++;  
    }  
}  
  
public void inserirEsq(int x){
```

```

        if(esq + dir >= array.length())
            MyIO.println("Erro ao inserir: deque cheio!");

        else{
            for(int i = 0; i < esq-1; i++)
                array[i+1] = array[i];

            array[0] = x;
            esq++;
        }
    }

    public int removerDir(){
        int saida = 0;

        if(this.dir == array.length()-1)
            MyIO.println("Erro ao remover: deque vazio!");

        else{
            saida = array[array.length()-1];

            for(int i = array.length(); i > array.length() - dir; i--)
                array[i-1] = array[i];

            dir--;
        }

        return saida;
    }

    public int removerEsq(){
        int saida = 0;

        if(this.esq == 0)
            MyIO.println("Erro ao remover: deque vazio!");

        else{
            saida = array[0];

            for(int i = 0; i < esq-1; i++)
                array[i] = array[i+1];

            esq--;
        }
    }

```

```

    return saida;
}

```

3 Questão 3

```

public int maiorHeap(int pos){
    int maior;

    if(pos *2 +1 <= n){
        maior = Math.max(maiorHeap(pos*2), maiorHeap(pos*2+1));
    }
    else if(pos *2 == n) maior = array[pos*2];
    else array[pos];

    return maior;
}

public int maiorHeapEsq(){
    return maiorHeap(1);
}

```

4 Questão 4

```

//Comparar duas Strings dentro de um time
// 0 -> Strings iguais
// 1 -> String 1 vem primeiro no alfabeto
// 2 -> String 2 vem primeiro no alfabeto
public int comparar(Time time1, Time time2){
    return compararStrings(time1.nome, time2.nome, 0);
}

private int compararStrings(String a, String b, int pos){
    int resp = 0;

    if(pos == a.length()-1 || pos == b.length()-1);
    else if(a.charAt(pos) < b.charAt(pos)) resp = 1;
    else if(a.charAt(pos) > b.charAt(pos)) resp = 2;
    else resp = compararStrings(a, b, pos+1);

    return resp;
}

```