Primeira Lista de Exercícios de PAA - 2019.1
Profs. Vinícius Fernandes dos Santos e Sebastián Urrutia

1. Leia os capítulos 1 a 5 e o capítulo 17 do Cormen.

2. Considere o algoritmo de ordenação descrito abaixo. Esse algoritmo recebe um vetor de números inteiros $A[1..n]$ distribuídos aleatoriamente e imprime os números desse vetor em ordem crescente.

```
SelectionSort(int A[1..n]) {
    for(i=1; i<n; i++)
        midx = findMinimum(A, i);
        swap(A[midx], A[i]);
    print(A);
}

findMinimum(int A[1..n], int idx) {
    min = idx;
    for(j=idx+1; j<=n; j++)
        if(A[min] > A[j])
            min = j;
    return min;
}
```

   (a) Utilizando invariantes de loop, mostre que esse algoritmo funciona.

   (b) Qual é a função de complexidade do número de comparações de elementos no melhor e pior caso?

3. Considere o algoritmo iterativo mostrado abaixo que encontra o maior e o menor elemento em um vetor $A[1..n]$. Considere ainda que os n elementos estão distribuídos aleatoriamente no vetor.

```
MaxMin(int A[1..n]){
    max = A[1];
    min = A[1];
    for (i=2; i<=n; i++)
        if (A[i]>max) max = A[i];
        else if(A[i]<min) min = A[i];
    print(min,max)
}
```

   (a) Qual é a função de complexidade do número de comparações de elementos no melhor e pior caso?

   (b) Utilizando análise probabilística, compute o número de comparações de elementos do vetor que serão realizadas no caso médio.

   (c) Implemente um algoritmo recursivo MaxMinRec usando o paradigma "dividir para conquistar" para resolver esse mesmo problema. (dica: modifique o mergesort). A complexidade do seu algoritmo deve ser inferior a $n\log(n)$.

   (d) Qual é a complexidade do seu algoritmo? Para isso, determine e resolva a sua equação de recorrência.

4. Sejam $f(n)$, $g(n)$ duas funções assintóticas positivas. Prove que as afirmativas abaixo são verdadeiras ou falsas, usando para isso as definições das notações assintóticas ou contra exemplos.

   (a) A relação Θ é simétrica, ou seja $g(n) = \Theta(f(n))$ se somente se $f(n) = \Theta(g(n))$.

   (b) $f(n) = O(f(n/2))$

   (c) $(n+a)^b = \Theta(n^b)$

   (d) $\Omega(f(n) + g(n)) = max(f(n), g(n))$

   (e) Considerando que $f(n) = \omega(g(n))$ então $f(n) + g(n) = \Omega(g(n))$

5. Encontre um limite assintótico firme para as equações de recorrência abaixo. Para cada uma delas, você deve encontrar a complexidade e demonstrar que ela está correta utilizando o método da substituição. Em seguida, verifique que seu resultado está correto usando o teorema mestre. Considere $T(1) = 1$.

   (a) $T(n) = T(n/2) + 1$

(b) $T(n) = 4T(n/2) + n$

(c) $T(n) = T(9n/10) + n$

6. Considere o seguinte problema: dado um vetor $A[1..n]$, queremos determinar a maior diferença $d = A[i] - A[j]$ tal que d seja par, para todos os possíveis valores de i e j.

   (a) Escreva um algoritmo para resolver o problema acima, sem ordená-lo;

   (b) Analise a complexidade do seu algoritmo. Forneça um limite justo, utilizando a notação Θ;

   (c) Assumindo que o vetor está ordenado inicialmente, é possível resolver este problema de forma mais eficiente. Forneça um algoritmo de tempo linear para resolver este problema e mostre que ele está correto.

Exercícios Extras (Possivelmente mais difíceis, para ajudar na prepapração para a prova, mas que não devem ser entregues)

1. Resolva a recorrência $T(n) = 3T(\sqrt{n}) + \log n$.

2. Mostre como ordenar n inteiros no intervalo $[1, k]$ em tempo linear $O(n + k)$.

3. Mostre como ordenar n inteiros no intervalo $[1, n^2]$ em tempo linear $O(n)$.

4. Mostre que para fazer o merge de duas listas com n elementos é necessário realizar pelo menos $2n - 1$ comparações no pior caso.

5. Seja uma matriz quadrada A com n^2 números inteiros que satisfaz as seguintes propriedades:

   - $A[i, j] \leq A[i + 1, j]$ para $1 \leq i \leq n - 1$ e $1 \leq j \leq n$;
   - $A[i, j] \leq A[i, j + 1]$ para $1 \leq i \leq n$ e $1 \leq j \leq n - 1$;

   Dado um elemento x, descreva um procedimento eficiente para determinar se x pertence a A ou não. Analise a complexidade do algoritmo proposto. Mostre que este problema tem complexidade $\Theta(n)$.