Universidade Federal Fluminense
Disciplina: Análise e Projeto de Algoritmos
Professor: Luís Felipe

Revisão AP 1

1. Considere o problema das ovais: n ovais são desenhadas no plano; quaisquer duas ovais se interceptam em exatamente 2 pontos; três ovais não se interceptam em um único ponto. Determine a relação de recorrência do número de regiões nas quais essas n ovais dividem o plano e obtenha sua fórmula fechada.

2. Obtenha a fórmula fechada da seguinte relação de recorrência: $a_n = 4a_{n-1} - 4a_{n-2} + n$, onde $a_0 = 4$ e $a_1 = 7$.

3. Considerando o seguinte algoritmo de busca, obtenha sua complexidade de caso médio, assumindo que a probabilidade de sucesso é p e é igualmente distribuída dentre as n posições da entrada.

```
i = 1
L[n + 1] = x
Enquanto L[i] ≠ x faça:
    i = i + 1
Se i ≠ n + 1 então:
    Busca = i
Senão
    Busca = 0
```

4. Seja f uma função que é uma soma de 4 termos, cujo termo de maior grau é um monômio de ordem 4. Mostre que $f = O(n^4)$. Para isso, obtenha valores de c e n_0 que justifique. Essa sua justificativa se aplicaria analogamente caso f fosse uma função igual a uma soma de k termos, cujo termo de maior grau fosse um monômio de ordem k? Justifique.

5. Dado um tabuleiro de dimensões $2 \times n$, determine de quantas formas ele pode ser completamente preenchido usando apenas as seguintes peças:

   - Dominós 2×1: podem ser colocados verticalmente ou horizontalmente.
   - Quadrados 2×2: cobrem exatamente duas colunas e duas linhas.

   Obtenha a relação de recorrência para calcular o número total de maneiras possíveis de ladrilhar um tabuleiro de tamanho $2 \times n$. Obs.: Não pode simplesmente escrever a relação de recorrência, mas deve justificar essa fórmula obtida.

   Resposta (recursividade sem a base):

   $T(n) = 2T(n-2) + T(n-1)$

6. Dados n amigos, queremos montar configurações distintas onde cada um pode permanecer sem par ou pode ser pareado com algum outro amigo, formando uma dupla. Em cada configuração, cada amigo só pode participar de uma dupla por vez. Quantas configurações distintas temos, dados n amigos? Escreva a relação de recorrência que expressa esse número.

   Resposta (recursividade sem a base):

   $f(n) = f(n-1) + (n-1)f(n-2)$

7. Determine a relação de recorrência que fornece o número de sequências quaternárias de n dígitos que contém um número par de zeros.

   Resposta: Seja a_n o número de sequências quaternárias com n dígitos que contém um número par de zeros. Considere as sequências com $n-1$ dígitos com número par de zeros. Para o n-ésimo dígito temos 3 opções, a saber: $1, 2, 3$. Assim, temos $3a_{n-1}$ sequências com n dígitos e número par de zeros que não terminam em 0. Entretanto, a sequência pode terminar em 0. Neste caso, a sequência com $n-1$ dígitos deve conter um número ímpar de zeros. O número de sequências com $n-1$ dígitos e número ímpar de zeros pode ser calculado utilizando a noção de complemento: total de sequências quaternárias com $n-1$ dígitos subtraída a quantidade de sequências quaternárias com $n-1$ dígitos e número par de zeros, i.e., $4^{n-1} - a_{n-1}$. Como os dois casos são disjuntos, temos, pelo princípio aditivo que, $a_n = 3a_{n-1} + 4^{n-1} - a_{n-1} = 2a_{n-1} + 4^{n-1}$, $a_1 = 3$, pois ao considerarmos sequências com um dígito, temos 3 possíveis.

8. Considere uma busca não ordenada em uma lista de n chaves distintas, onde n é múltiplo de 3. As chaves são divididas em dois grupos:

   - $\frac{2n}{3}$ primeiras chaves são letras;
   - $\frac{n}{3}$ demais chaves são números.

   As chaves associadas as letras estão organizadas de forma arbitrária na lista, assim como as chaves associadas aos números. Para resolver esse problema, considere o seguinte Algoritmo 1.

Algorithm 1 BuscaChave(lista, chave, n)

```
Require: lista: vetor de n chaves distintas
Require: chave: chave a ser buscada
Require: n: número de chaves (múltiplo de 3)
if chave é uma letra then
   for i = 1 to 2n/3 do
      if lista[i] = chave then
         return "Chave encontrada na posição", i
      end if
   end for
   return "Chave não está na lista"
else if chave é um número then
   for i = 2n/3 + 1 to n do
      if lista[i] = chave then
         return "Chave encontrada na posição", i
      end if
   end for
   return "Chave não está na lista"
else
   return "Tipo de chave inválido"
end if
```

Suponha que:

- A probabilidade de busca de uma letra é igual entre todas as letras;
- A probabilidade de busca de um número é o dobro da de uma letra;
- A soma total das probabilidades de todas as chaves é q, com $0 < q < 1$.

Determine uma expressão para a complexidade média da busca nesse caso, assumindo que em tempo constante sabemos se a chave a ser buscada é uma letra ou número.

Resposta: Seja p a probabilidade de busca de uma chave letra. Então, a probabilidade de busca de uma chave número é $2p$.

Como há $\frac{2n}{3}$ letras e $\frac{n}{3}$ números, temos:

$$\left(\frac{2n}{3}\right) p + \left(\frac{n}{3}\right)(2p) = q$$

$$\frac{2np}{3} + \frac{2np}{3} = \frac{4np}{3} = q \Rightarrow p = \frac{3q}{4n}$$

Assim, a probabilidade de uma letra é $\frac{3q}{4n}$ e de um número é $\frac{6q}{4n} = \frac{3q}{2n}$.

Agora, sabendo que as letras ocupam as primeiras $\frac{2n}{3}$ posições e os números as últimas $\frac{n}{3}$ posições, façamos a partição das entradas associadas as letras sendo da seguinte forma:

(a) E_i, para $1 \leq i \leq \frac{2n}{3}$ são as entradas que retornam sucesso para busca associada a uma letra. Neste caso, o algoritmo realizou i passos;

(b) E_i, para $\frac{2n}{3} + 1 \leq i \leq n$ são as entradas que retornam sucesso para busca associada a um número. Neste caso, o algoritmo realizou $j = i - \frac{2n}{3}$ passos, dado que sabendo que a entrada é um número, então o algoritmo iniciará já na $(\frac{2n}{3} + 1)$-ésima posição;

(c) $E'_{\frac{2n}{3}}$ e E'_n são as entradas que retornam fracasso para quando é uma letra ou número que não esteja na lista, respectivamente. Neste caso, tanto para uma letra quanto para um número, o algoritmo precisou realizar a busca até a $\frac{2n}{3}$-ésima e n-ésima posição, respectivamente. Tendo assim, $\frac{2n}{3}$ e $\frac{n}{3}$ operações. Ademais, como $1 - q$ é a probabilidade de fracasso, $\frac{1-q}{2}$ é a probabilidade do fracasso ser uma letra ou um número.

Assim, a complexidade média será:

$$C.M. = \sum_{i=1}^{2n/3} i \cdot \frac{3q}{4n} + \sum_{j=1}^{n/3} j \cdot \frac{3q}{2n} + \frac{2n}{3}\frac{(1-q)}{2} + \frac{n}{3}\frac{(1-q)}{2}$$