Prova 1 - 2021-1

1- Considere o seguinte algoritmo.

```
Girassol (A)
  n = A.length
  M = A[1] ;
  for j = 2 to n do
      M = (M*(j-1) + A[j])/j ;
  return M;
```

a) O que este algoritmo realiza na lista A?

$A = [2, 5, 8, 5]$

$j = 2, 3, 4$

$M = 2, 3.5, 5, 5$

Calcula a média dos elementos do vetor A.

b) Para demonstrar que este algoritmo realmente funciona, pode-se utilizar a técnica de indução matemática. Para tal, qual hipótese indutiva, ou loop invariante você utilizaria?

Ao início da iteração j do for, M possui a média do subvetor $A[1..j-1]$

c) Demonstre a parte da manutenção do loop invariante da letra b

1-(inicialização) $j = 2$, subvetor $A[1]$ e $m = A[1]$, que é a média de um único elemento

2-(manutenção) Assumindo que o valor de M é igual à média dos elementos de $A[1..j-1]$, $M = \frac{\sum_1^{j-1} A[i]}{j-1}$. Portanto, na iteração j, o que fazemos é

$$\frac{M \cdot j-1 + A[j]}{j} = \frac{\frac{\sum_1^{j-1} A[i]}{j-1} \cdot j-1 + A[j]}{j} = \frac{\sum_{i=1}^{j} A[i]}{j}$$

3-(terminação). No fim $j = n+1$, logo temos a média dos valores de $A[1..n]$ armazenado em M.

2. Resolva:

a) Defina uma função $f: \mathbb{N} \to \mathbb{N}$ tal que $f(2n) \in O(f(n))$. Demonstre que esta função satisfaz a propriedade pedida.

Função identidade. Neste caso, $\boxed{f(2n) = 2 \cdot f(n) \leq c \cdot f(n) \\ = 2n \leq c \cdot n}$

para $c \geq 2$ e n_0 1.

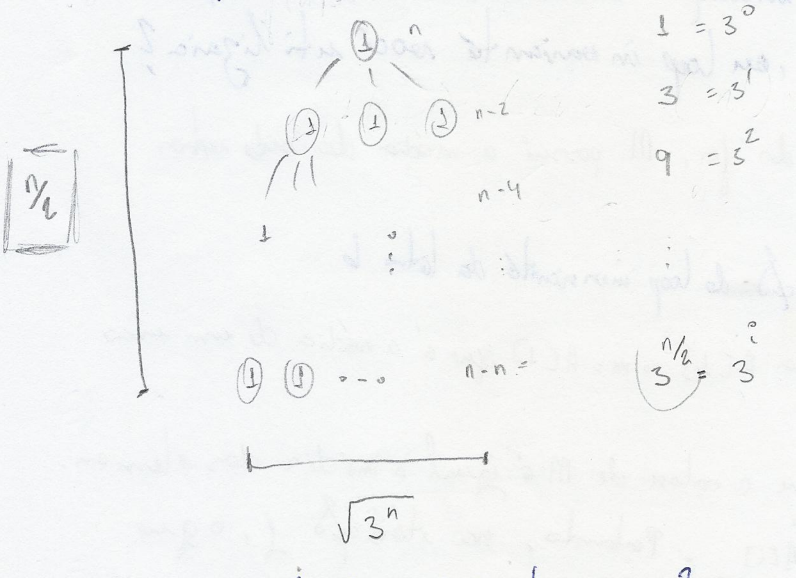
b) Defina uma função $f: \mathbb{N} \to \mathbb{N}$ tal que $f(2n) \notin O(f(n))$. Demonstre que esta função satisfaz a propriedade pedida.

$F(n) = n!$. Assumindo que $f(2n) \in O(f(n))$, precisamos provar

$f(2n) \leq c \cdot f(n) \Rightarrow (2n)! \leq c \cdot n!$, o que é um absurdo para qualquer $c \geq 0$. Portanto, $f(2n) \notin O(f(n))$

c) Seja $S(n)$ uma função definida recursivamente, com casos base $S(1) = S(2) = 1$ e recorrência $S(n) = 3S(n-2) + 1$

(i) Esboce a árvore de recorrência desta função.



(ii) Quantos níveis ela possui?

$n/2$

(iii) Qual o custo do k-ésimo nível?

3^K

(iv) Encontre uma função $f(n)$, definida explicitamente, tal que $S(n) \in O(f(n))$ (mas não precisa demonstrar por indução que funciona).

profundidade $= n/2$

número de nós por nível $= 3^i$ custo por nó $= 1$

$$S(n) = \sum_{i=1}^{n/2} 3^i \cdot 1 = 3^{n/2+1} - 1 = \Theta(3^n) = \Theta(3^{n/2})$$

substituindo $S(n) \leq c \cdot 3^n$

$S(n) = 3S(n-2) + 1$

$\leq 3c \cdot 3^{(n-2)} + 1$

$= c\, 3^{n-1} + 1$

$\leq c\, 3^n$

$S(n) \leq c \cdot 3^{n/2} - d$

$S(n) = 3S(n-2) + 1$

$\leq 3c \cdot 3^{(n-2)/2} - 3d + 1$

$= c \cdot 3^{(n-2)/2+1} - 3d + 1$

$= c \cdot 3^{n/2} - 3d + 1$

$\leq c \cdot 3^{n/2} - d$ p/ $d \geq 1$

3. Uma permutação de uma lista com n elementos foi gerada de modo uniformemente aleatório. O objetivo desta questão é determinar a quantidade esperada de elementos que terminam em uma posição diferente da original.

a) Seja X_i a variável indicadora de que o elemento na posição i ficou em posição diferente de i. Qual o valor esperado de X_i?

$E[X_i] = \frac{n-1}{n}$

b) Seja X a variável aleatória que conta quantos elementos terminaram em posições diferentes das originais. Qual relação entre X e os X_i?

$X = \sum X_i$

c) Quantos elementos é esperado terem terminado em posição diferente?

$E[X] = \sum E[X_i] \Rightarrow \sum_{i=1}^{n} \frac{n-1}{n} = n \cdot \frac{n-1}{n} = n-1$ elementos

4. Suponha que realizamos uma sequência de n operações numa estrutura de dados, onde a k-ésima operação custa $2k$ se k é potência de 2, e custa 1 caso contrário. Seja $T(n)$ o custo das operações realizadas.

a) Dado n, sabemos que há apenas uma única potência de 2 entre $\lfloor n/2 \rfloor + 1$ e n. Ache $f(n) \in \Theta(n)$, tal que $T(n) \leq T(n/2) + f(n)$.

Resolvendo "na mão"

$$T(n) = \sum_{i=1}^{\lg n} 2 \cdot 2^i + \sum_{i=1}^{n-\lg n} 1 = 2 \cdot \sum 2^i + n - \lg n = 2 \cdot (2^{\lg n+1} - 1) + n - \lg n =$$

$$2 \cdot 2 \cdot 2^{\lg n} - 2 + n - \lg n = 4n - 2 + n - \lg n = 5n - \lg n - 2 = \Theta(n)$$

b) Verifique por indução que $T(n) \leq 5n$ (se você tiver dificuldade, pode ser necessário melhorar sua escolha da $f(n)$).

a) Entre $n/2$ e n temos $\sum_{n/2}^{n-1} 1 + 2k = n/2 - 1 + 2k \leq n/2 - 1 + 2n = \boxed{\frac{5n}{2} - 1}$

onde $n/2 \leq k \leq n$.

$T(n) \leq 5n$:

$$\begin{aligned} T(n) &\leq T(n/2) + 5n/2 - 1 \\ &\leq 5n/2 + 5n/2 - 1 \\ &= 5n - 1 \\ &\leq 5n \end{aligned}$$

c) Em uma análise de custo amortizada, utilizando o método da contabilidade, desejamos atribuir um custo fictício para cada operação de modo que o custo total real, em qualquer momento, seja inferior ao custo fictício acumulado. Qual seria uma escolha razoável para o custo fictício de cada operação?

Como visto anteriormente, entre $n/2$ e n temos no máximo $5n/2 - 1$ operações. Dividindo pelas $n/2$ operações no intervalo, temos $(5n/2 - 1)/(n/2) = 5 - 2/n$ que é ≤ 5. Logo, o custo de cada operação é igual a 5.