

# Lista de Exercícios – Análise de Algoritmos e Notação Assintótica

## Introdução à Ciência de Computação II

Prof. Moacir A. Ponti

6 de novembro de 2020

1. Resolva as relações de recorrência dadas pelas funções abaixo e respectivos casos base/triviais (condições de parada):
  - $T(n) = 7 + T(n - 1)$ , com  $T(0) = 0$ ;
  - $T(n) = c + T(n - 3)$ , com  $T(p) = 1, p \leq 3$ ;
  - $T(n) = n/2 + T(n - 1)$ , com  $T(1) = 1$ ,
  - $T(n) = c + 2 \cdot T(n/2)$ , com  $T(1) = 1$ ,
  - $T(n) = c + T(n/4) + T(n/2)$ , com  $T(1) = 1$ ,
2. Considere o código abaixo, sendo operações relevantes as atribuições e comparações.
  - a) Encontre a função  $g(n)$  que representa o número de operações realizadas pela função 'preenche',
  - b) Encontre uma relação de recorrência  $f(n)$  para a função 'menores', em termos da execução do laço mais externo, trocando essa repetição por uma recursão. Preferencialmente, resolva primeiro o laço mais interno, incluindo-o na equação do laço externo.
  - c) Encontre a forma fechada para  $f(n)$ .
  - d) Escreva a função  $T(n)$  incluindo todas as operações da função 'main()'. A partir de  $T(n)$ , encontre a função de eficiência assintótica usando as notações  $O$ ,  $\Omega$  e  $\Theta$ . Use a definição formal e encontre as constantes.

```
void preenche(int *v, int N) {
    int i;
    for (i = 0; i < N; i++) {
        *(values+i) = rand()%1000;
    }
}

void menores(int *v, int* lt, int N) {
    for (i = 0; i < N; i++) {
        for (j = 0; j <= i; j++) {
            if (v[j] < v[i])
                lt[i]++;
        }
    }
}
```

```

    }
}
int main (void) {
    srand(NULL);
    int N;
    scanf("%d", &N);
    int *values = malloc(N*sizeof(int));
    int *lessthan = calloc(N,sizeof(int));
    int i,j;
    preenche(values,N);
    menores(values,lessthan, N);
    free(values);
    free(lessthan);
    return 0;
}

```

3. A partir das funções de contagem de operações abaixo, encontre a função de eficiência usando a notação  $\Theta$ .

- $f(n) = 5n - \log_2 n$ ;
- $f(n) = \frac{15n}{2} + 15$
- $f(n) = \frac{n(n-2)}{3} - 5$

4. Verifique se as seguintes proposições estão corretas

- 7 é  $O(n)$ ;
- $n$  é  $O(1)$ ;
- $n$  é  $\Omega(1)$ ;
- $2n^4 - n + 1$  é  $O(n^4)$ ;
- $100n^4 + n^3$  é  $O(2^n)$ ;
- $n^n$  é  $O(2^n)$ ;

5. Encontre a função de contagem de operações  $f(n)$  para os seguintes somatórios:

- $\sum_{i=0}^{n-1} (i^2 + 1)^2$
- $\sum_{i=2}^{n-1} (\log_2 i^2)$
- $\sum_{i=1}^n (i + 1) \cdot 2^{i-1}$
- $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i + j)$

6. Considere um computador com *clock* de 2GHz, que realiza cada operação relevante em 1 ciclo. Estime, apenas com esses dados, o tempo necessário para que ele execute os algoritmos cujas funções de eficiência estão abaixo, considerando que os algoritmos processam números do tipo `float`, e que o tamanho dos dados de entrada é de 140MB:

- $f(n) = 5n - \log_2 n$
- $f(n) = \frac{n^2 - 2n}{2}$
- $f(n) = \frac{n^3 + n}{2}$
- $f(n) = n \cdot 2^n$

7. Considere o código abaixo, sendo operações relevantes apenas a constante  $a$

- a) Encontre a função que representa o número de operações realizadas pela função 'teste' por meio de uma equação de recorrência. Para isso obtenha relações de recorrência para cada laço (while, for), e substitua na equação.
- b) Encontre a forma fechada para a função.
- c) A partir da forma fechada, demonstre a função de eficiência assintótica usando as notações  $O$ ,  $\Omega$  e  $\Theta$ . Use a definição formal e encontre as constantes.

```
void teste(int n) {
    int m = n;
    for (i = n; i > 0; i = i/2) {
        for (j = i; j < m; j++) {
            // 'a' operacoes
        }
    }
    while (m > 0) {
        // 'a' operacoes
        m = m * 0.25;
    }
}
```