

Relatorio de Laboratório de Introdução à Ciência da Computação II – Avaliativo 3

Leonardo Kenzo Tanaka e Pedro Teidi de Sá Yamacita

13 de setembro de 2025

1 Análise do Código Iterativo

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int tamanho_nome(char *nome)
5 {
6     int tamanho = 0;
7     for (int i = 0; nome[i] != '\0'; i++)
8     {
9         // pular espacos e '-' que adicionamos no caso do nome ser composto
10        if (nome[i] != ' ' && nome[i] != '-')
11            tamanho++;
12    }
13    return tamanho;
14 }
```

1.1 Análise da versão Iterativa

O código acima apresenta a forma iterativa de contar a quantidade de caracteres de um nome simples ou composto com até duas palavras devido a como foi feita a manipulação da string 'linha' na main.

A contagem de caracteres foi feita de uma forma bem simples por meio de um "for" que itera até encontrar o final da string 'nome' determinado pelo "\0", adicionando ao tamanho sempre que o caracter apontado seja diferente de espaço ou de '-' que foi adicionado na manipulação da linha.

Em decorrência do uso de "fors", a complexidade do programa assume o valor de $O(n)$, sendo n o número máximo de iterações que um dos loops terá de fazer (os prints também são feitos com o uso de "fors").

2 Análise do Código Recursivo

```
1 #include <stdio.h>
2 #include <ctype.h>
3 #include <string.h>
4
5 //Funcao recursiva que retorna a quantidade de letras do nome
6 int ContadorLetras(char *nome, int quantLetras){
7
8     //Passa por todos os caracteres da string
9     char letra = nome[0];
10
11     //Verificacoes e contador de letras
12     if(!isalpha(letra) && letra != '-'){
13         nome++;
14     }
15     if(isalpha(letra)){
16         quantLetras++;
17         nome++;
18     }
19 }
```

```
19     if(letra == '-'){
20         return quantLetras;
21     }
22
23     //Chamada recursiva
24     ContadorLetras(nome, quantLetras);
25 }
```

2.1 Análise do Código Recursivo

O código acima apresenta a versão recursiva para contar a quantidade de caracteres de um nome, ignorando os espaços.

A função recursiva de contagem começa analisando o primeiro caractere da string fornecida e incrementa o contador em 1 caso seja uma letra. Se for um espaço, ignora e passa para o próximo caractere; se for o símbolo “-”, que indica o término do nome, retorna o contador com o resultado final. Enquanto o caractere for diferente de “-”, a função chama a si mesma recursivamente até obter o resultado.

Dessa forma, é possível perceber que a complexidade desse código é $O(n)$, pois ele percorre todos os caracteres da string. Além disso, esse método é mais complexo e desnecessário, já que também percorre toda a string original, mas faz várias chamadas de função sem necessidade.

3 Comparação Entre os Dois Códigos

É possível perceber que ambos os códigos realizam a mesma função, utilizando uma lógica semelhante de analisar caractere por caractere da string para contar a quantidade de letras de cada nome.

No entanto, o código iterativo é muito mais simples, intuitivo e de fácil manutenção, pois adota uma abordagem direta e objetiva. Em contrapartida, o código recursivo é mais complexo e emprega um método excessivamente elaborado para resolver um problema simples, como a contagem de caracteres.

Além disso, ambos os códigos possuem a mesma complexidade de $O(n)$, já que precisam percorrer todos os caracteres da string. Portanto, o código mais eficiente e simples é o iterativo.