Somatórios e relação de recorrência

Matemática Discreta

Prof. Lucas Ismaily

1º Semestre de 2022

Aluno: [ ] Matrícula: [ ]

Questões:

1. Escreva as seguintes soma na notação de somatório:

   (a) $2 * 1 + 2 * 2 + \ldots + 2 * n$.

   (b) $1 + 3 + 5 + 7 + \ldots + (2n - 1)$.

   (c) $\frac{1}{3*4} + \frac{1}{5*6} + \frac{1}{7*8} + \ldots + \frac{1}{n(n+1)}$.

   (d) $\frac{1}{3*5} + \frac{1}{5*7} + \ldots + \frac{1}{(2n-1)(2n+1)}$.

   (e) $4 + 8 + \ldots + 2^n$.

   (f) $a + a^2 + a^3 + \ldots + a^n$.

   (g) $1a + 2a^2 + 3a^3 + \ldots + na^n$.

2. Sabendo que

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$
$$\sum_{i=1}^{n} a^i = \frac{a^{n+1} - a}{a - 1}$$

   Utilizando as propriedades dos somatórios, encontre a fórmula fechada para os seguintes somatórios:

   (a) $1 + 4 + 7 + \ldots + (3n - 2)$.

   (b) $1 + 3 + 5 + \ldots + (2n - 1)$.

   (c) $-1 + 6 + 25 + \ldots + (3^n + n - 5)$.

3. Encontre a fórmula fechada para os seguintes somatórios:

   (a) $\sum_{k=1}^{n} k$.

   (b) $\sum_{k=1}^{n} 2^k$.

   (c) $\sum_{k=1}^{n} \sum_{j=1}^{m} kj$.

4. Sabendo que

$$(n+1)^3 - n^3 = 3n^2 + 3n + 1 \tag{1}$$

   (a) Usando a igualdade anterior, encontre uma fórmula para $\sum_{k=1}^{n} 3k^2 + 3k + 1$.

   (b) Usando o item anterior, encontre uma fórmula para $\sum_{k=1}^{n} k^2$.

5. Sabendo que $\dfrac{1}{k(k+1)} = \dfrac{1}{k} - \dfrac{1}{k+1}$. Encontre uma fórmula fechada para $\sum_{k=1}^{n} \frac{1}{k(k+1)}$.

6. Considere o seguinte código C.

```c
#include<stdio.h>


void prog1(int n){
        if (n==1){
                printf("oi");
                printf("oi");
                printf("oi");
        }
        else{
                for (int i=1; i<=5; i++){
                        printf("oi");
                }
                prog1(n-1);
        }
}

int main(){
  int n;
  scanf("%d", &n);
  prog1(n);
  return 0;
}
```

   Seja $T(n)$ o número de vezes que a palavra *oi* é impressa quando *prog1* é chamada com o parâmetro n.

(a) Calcule o valor de $T(1)$.

(b) Determine uma relação recorrência para $T(n)$.

(c) Resolva a relação de recorrência pelo método iterativo.

7. Considere o seguinte código em C.

```c
#include<stdio.h>


void prog1(int n){
        if (n==1){
                printf("oi");
                printf("oi");
                printf("oi");
                printf("oi");
                printf("oi");

        }
        else{
                for (int i=1; i<=n+3; i++){
                        printf("oi");
                }
                prog1(n-1);
        }
}

int main(){
  int n;
  scanf("%d", &n);
  prog1(n);
  return 0;
}
```

Seja $T(n)$ o número de vezes que a palavra *oi* é impressa quando *prog1* é chamada com o parâmetro n.

(a) Calcule o valor de $T(1)$.

(b) Calcule o valor de $T(2)$.

(c) Determine uma relação recorrência para $T(n)$.

(d) Resolva a relação de recorrência pelo método iterativo.

8. A sequência de Fibonacci pode ser definida da seguinte maneira:

$$\begin{aligned} f_1 &= 1 \\ f_2 &= 1 \\ f_n &= f_{n-1} + f_{n-2}, \forall n \geq 3 \end{aligned}$$

Mostre que a sequência de Fibonacci satisfaz à seguinte identidade: $f_1+f_2+\ldots+f_n = f_{n+2} - 1$. Dica: Substitua cada termo f_i por $f_{i+2} - f_{i+1}$.

9. Considere o seguinte código em C.

```c
#include<stdio.h>


void prog1(int n){
        if (n==1){
                printf("oi");
                printf("oi");

        }
        else{
                for (int i=1; i<=3; i++){
                        printf("oi");
                }
                prog1(n-1);
                prog1(n-1);
        }
}

int main(){
  int n;
  scanf("%d", &n);
  prog1(n);
  return 0;
}
```

Seja $T(n)$ o número de vezes que a palavra *oi* é impressa quando *prog1* é chamada com o parâmetro n.

(a) Calcule o valor de $T(1)$.

(b) Calcule o valor de $T(2)$.

(c) Determine uma relação recorrência para $T(n)$.

(d) Resolva a relação de recorrência pelo método iterativo. Dica: lembre-se que $\sum_{k=0}^{n} ar^k = \frac{ar^{n+1} - a}{r - 1}$, para $r \neq 1$.