**Nome: Gustavo Hammerschmidt.**

**TDE 03 - Indução Matemática**

**Demonstre os seguintes exemplos:**

1. **2 + 6 + 10 + ... + (4n - 2) = 2n²**
2. **1 + 5 + 9 + ... + (4n – 3 ) = n(2n - 1)**
3. **4 + 10 + 16 + ... + (6n - 2) = n(3n + 1)**
4. **2^(n) < n! para n >= 4**
5. **1 + 2 + ... + n < n² para n > 1**
6. **Números triangulares T(n) = n(n + 1)/2**
7. **Prove que qualquer valor postal maior ou igual a 64 unidades monetárias pode ser obtido usando-se somente selos de 5 e 17.**
8. **Um quebra-cabeça é montado por junções sucessivas de peças que se organizam em blocos. Um movimento é feito cada vez que uma peça é adicionada a um bloco, ou quando dois blocos são agrupados. Prove que não importa como os movimentos são realizados, são necessários n-1 movimentos para montar um quebra-cabeça com n peças.**
9. **Considere um jogo em que dois jogadores se alternam para remover qualquer número positivo de cartas que eles pegam a partir de dois montes de cartas de baralho. O jogador que tirar a última carta ganha o jogo. l Mostre que se duas pilhas contém o mesmo número de cartas inicialmente, o segundo jogador sempre ganha o jogo, ou tem uma estratégia para isso.**

**Respostas:**

1. **2 + 6 + 10 + ... + (4n - 2) = 2n²**

P(1) = 2n² = 2. 1² = 2 <- Base

P(K) => 2 + 6+ 10 + ... + (4K - 2) = 2k²

P(K+1) => 2 + 6 + 10 + ... + (4K+4-2) = 2(K+1)²

P(K+1) => P(K) + (4K+4-2) = 2(K+1)²

P(K+1) => 2K² + 4K + 2 = 2(K + 1)²

Comprovado por Indução fraca.

1. **1 + 5 + 9 + ... + (4n – 3 ) = n(2n - 1)**

P(1) = n(2n - 1) = 1x(2x1 - 1) = 1 <- Base

P(k) => 1 + 5 + 9 + ... + (4k - 3) = k(2k - 1)

P(k + 1) => 1 + 5 + 9 + ... + (4x(k+1) - 3) = (k+1)x(2x(k+1) - 1)

P(k + 1) => P(k) + (4x(k+1) - 3) = (k+1)x(2x(k+1) - 1)

P(k + 1) => k(2k - 1) + (4x(k+1) - 3) = (k+1)x(2x(k+1) - 1)

P(k + 1) => 2k² + 3k + 1 = 2k² + 3k + 1

Comprovado por Indução fraca.

1. **4 + 10 + 16 + ... + (6n - 2) = n(3n + 1)**

P(1) = n(3n + 1) = 1(3x1 + 1) = 4 <- Base

P(k) => 4 + 10 + 16 + ... + (6k - 2) = k(3k + 1)

P(K + 1) => 4 + 10 + 16 + ... + (6k + 6 - 2) = (k + 1)(3(k+1) + 1)

P(K + 1) => P(k)+ (6k + 6 - 2) = (k + 1)(3(k+1) + 1)

P(K + 1) => k(3k + 1) + (6k + 6 - 2) = (k + 1)(3(k+1) + 1)

P(K + 1) => 3k² + 7k + 4 = 3k² + 7k + 4

Comprovado por Indução fraca.

1. **2^(n) < n! para n >= 4**

P(4) = 2^(4) < 4! => 16 < 24 (Verdade) <- Base

P(k) => 2^(k) < k!

P(k + 1) => 2^(k + 1) < (k + 1)!

P(k + 1) => 2 x P(k) < (k + 1)!

2 x P( k) => 2^(k + 1) < 2\*(k!)

P(k + 1) => 2^(k + 1) < (k + 1)!

P(k + 1) => 2\*(k!) < (k + 1)! Para n > 1

Comprovado por Indução fraca.

1. **1 + 2 + ... + n < n² para n > 1**

P(1) = 1 <- Base

P(k) => 1 + 2 + ... + k < k²

P(k+1) => 1 + 2+ … + k + (k+1) < (k + 1)²

k < k² => k + (k+1) < k² + (k+1) para k > 1

k + k + 1 < k² + k + 1 < k² + 2k + 1

k² + k + 1 < k² + 2k + 1 => k² + k + 1 < (k +1 )²

1. **Números triangulares T(n) = n(n + 1)/2**

T(1) = 1 x (1+1)/2 = 1 <- Base

T(k) => 1 + 3 + 6 + 10 + ... + k = k x (k+1)/2

T(k + 1) => 1 + 3+ 6 + 10 + ... + k + (k+1) = (k+1)x((k+1)+1)/2

T(k + 1) => { k x (k+1) / 2 + (k+1) = (k+1)(k+2)/2 } x 2

T(k + 1) => k x (k+1) + 2 x (k+1) = (k+1)(k+2)

T(k + 1) => k² + k + 2k + 2 = k² + 3k + 2

T(k + 1) => k² + 3k + 2 = k² + 3k + 2

Comprovado por Indução fraca.

1. **Demonstração:**

S(64) = 64 <- Base

Para 64 =< r =< k

64 = 6(5) + 2(17)

65 = 13(5) **(5 porque é o valor mínimo)**

66 = 3(5) + 3(17)

67 = 10(5) + 17

68 = 4(17)

K + 1 >= 69

K + 1 > = 64 + 5

(K + 1) – 5 >= 64 <- Provado que é possível obter selos maiores que 64 com selos de 5 e 17 entre a faixa de 64 =< n =< 69.

K + 1 >= 64

((K+1) – 5) + 5 >= 64+5

K+1 >= 69 <- provado que eu posso ir de selos de 64 a 69 com selos de 5 e 17.

Comprovado por Indução forte.

1. **Demonstração:**

Q(K + 1) => número de peças.

Metade de Q é igual a 1 =< r1 =< k e 1 =< r2 =< k

e r1+r2 = k.

O quebra-cabeça original tem (r1-1)+(r2-1) + 1 movimentos.

(r1-1)+(r2-1) + 1 => (r1+ r2) – 1

* (k + 1) – 1 = k movimentos.

Comprovado que, em um quebra-cabeça de tamanho k+1 , o número de movimentos é k.

1. **Demonstração:**

Pilha1 = Pilha2 = k + 1 para 1 =< r =< k .

Jogador 1 retira uma: k + 1 – r.

Jogador 2 faz o mesmo na outra pilha.

O escopo agora é 1 =< k +1 – r =< k

Depois de o jogador 1 tiver tirado k +1 cartas da pilha,

sobrará uma para o jogador 2 tirar.