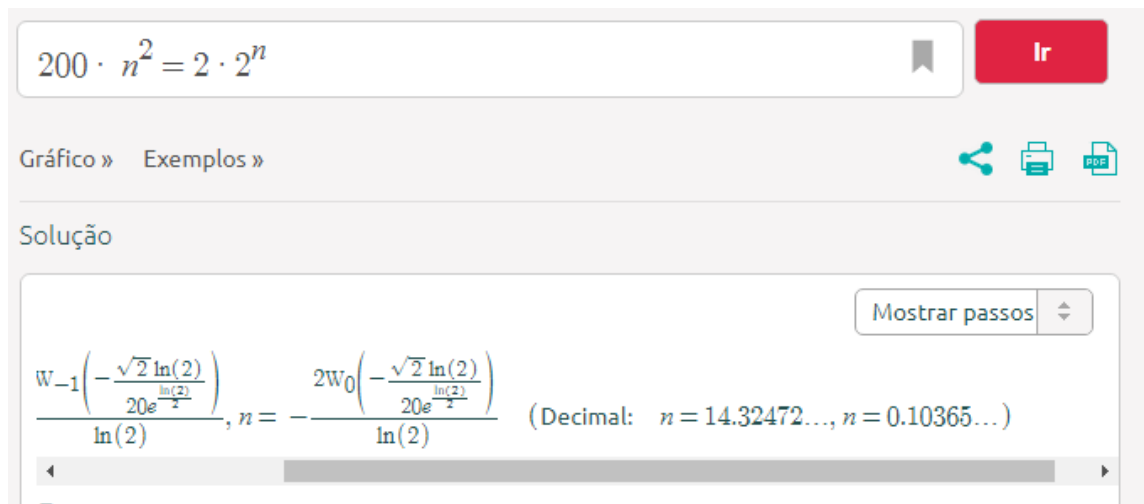


Exercícios Prova 1

1. Considere dois programas A e B com complexidade $200n^2$ e $2 \cdot 2^n$, respectivamente. Qual é o mais eficiente? (1,5)
2. Um algoritmo tem complexidade $2n^2$. Num certo computador, num tempo t , o algoritmo resolve um problema de tamanho x . Imagine agora que você tem disponível um computador 30 vezes mais rápido. Que parcela do tempo t precisará para resolver um problema 3 vezes maior? (1,5)
3. Escreva o pseudocódigo de um algoritmo que troque os valores contido em um arranjo A de n posições pela seguinte política: cada elemento i dentro do arranjo será substituído pela soma de todos os $(i-1)$ elementos mais o elemento i . Por exemplo, dado um arranjo [1; 2; 3; 4; 5] após a aplicação da função teríamos esse arranjo preenchido com os seguintes valores [1; 3; 6; 10; 15]. (2,0)
4. Qual invariante de laço esse algoritmo mantém? Usando um invariante de laço, prove que seu algoritmo é correto. Certifique-se de que seu invariante de laço satisfaz as três propriedades necessárias. (2,0)
5. Para esse algoritmo forneça os tempos de execução do melhor caso e do pior em notação Θ . (3,0)

Respostas

1.



Equation: $200 \cdot n^2 = 2 \cdot 2^n$

Gráfico » Exemplos »

Solução

Mostrar passos

$$\frac{W_{-1}\left(-\frac{\sqrt{2} \ln(2)}{20e^{\frac{\ln(2)}{2}}}\right)}{\ln(2)}, n = -\frac{2W_0\left(-\frac{\sqrt{2} \ln(2)}{20e^{\frac{\ln(2)}{2}}}\right)}{\ln(2)}$$
 (Decimal: $n = 14.32472\dots, n = 0.10365\dots$)

Entre os valores encontrados (0,10365 – 14,32472) o $(2 \cdot 2^n)$ é mais eficiente, e entre (0-0,10365) e acima de 14,32472 o $200 \cdot n^2$ é mais eficiente. Nos valores 0,10365 e 14,32472 eles são similares.

n	caso A	caso B
0	0	2
0,10365	2,148665	2,148977
1	200	4
2	800	8
3	1800	16
13	33800	16384
14	39200	32768
14,32472	41039,52	41039,34
15	45000	65536

2.

$$2 * n^2 = t$$

$$\frac{2 * (3n)^2}{30} = t'$$

$$\frac{9 * t}{30} = t'$$

$$\frac{3t}{10} = t'$$

Docente Manoel Ribeiro Filho

Discente Maximiliano Simon Borges Bogado 201840601026

Disciplina Complexidade de Algoritmos

Precisará de três décimos do tempo T

3.

Troca dos valores

1-For j=1 to A.comprimento

2- k = A[j]

3- i = j-1

4- novoValor = novoValor + k

5- A[j] = novoValor

```
#include <stdio.h>
int main(){
    int a[6] = {1,2,3,4,5,6};
    int i,j,k;
    int soma=0;

    for (j=0; j<6; j++){
        k = a[j];
        i = j-1;
        novoValor = novoValor+k;
        a[j]=novoValor;
    }
    for (i=0; i<6; i++){
        printf("%d ",a[i]);
    }
    return 0;
}
```

4. O invariante de laço seria a linha 3 do pseudocódigo, o subarranjo (i=j-1), são os itens que foram trocados de valor
5. Para este tipo de problema, o melhor caso seria que todos os elementos do arranjo serem 0, colocando uma verificação para pular para o próximo item. Mas de todo modo o algoritmo deve percorrer todo o arranjo.

$$T(n) = c_1 n + c_2 n + c_6 \sum_{j=2}^n (t-1) + c_7 \sum_{j=2}^n (t-1)$$



Docente Manoel Ribeiro Filho

Discente Maximiliano Simon Borges Bogado 201840601026

Disciplina Complexidade de Algoritmos