Análise de Algoritmos
Conceitos e Notações
Exercícios

Nelson Cruz Sampaio Neto
nelsonneto@ufpa.br

Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Faculdade de Computação

8 de setembro de 2023

Questão 1

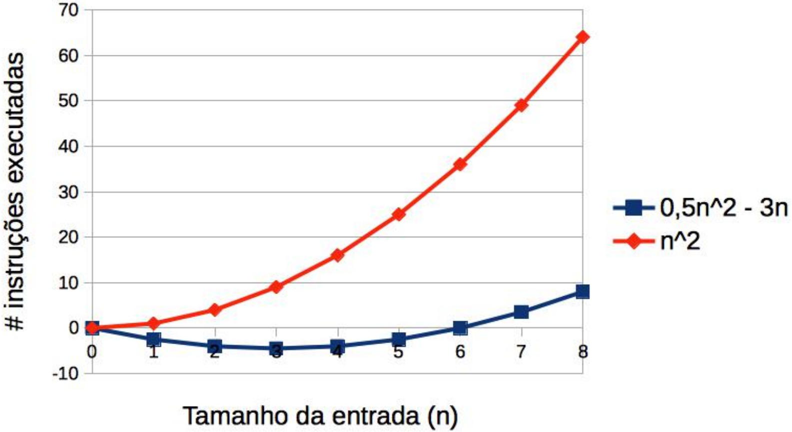
- Considere que $f(n) = 0,5n^2 - 3n$ e $g(n) = n^2$.

  Agora, usando a notação Big-O, mostre que $f(n)$ é $O(g(n))$:

  $$0 \leq f(n) \leq c\ g(n), \text{ sempre que } n \geq k.$$

- Fixando $c = 1$, tem-se:
  - Para $n < 6$: $f(n)$ é negativa e **não** atende a condição Big-O.
  - Para $n = 6$: $f(6) = 0$.
  - Para $n \geq 6$: atende Big-O com $0 \leq f(n) \leq g(n)$.
- Logo, $f(n)$ é $O(g(n))$, com $c = 1$ e $k = 6$.

Questão 1



70
60
50
40
30
20
10
0
-10
0
1
2
3
4
5
6
7
8
instruções executadas
Tamanho da entrada (n)
0,5n^2 - 3n
n^2

Questão 2

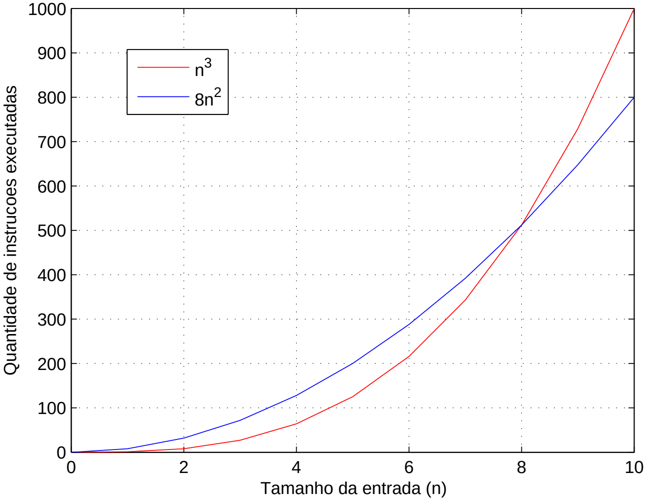
- Dado dois algoritmos A e B que resolvem o mesmo problema e possuem complexidade $8n^2$ e n^3, respectivamente. Qual é o maior valor de n para o qual o algoritmo B é mais eficiente que o algoritmo A?

  Até $n = 7$, B é mais eficiente que A.

- Qual é o algoritmo mais eficiente?

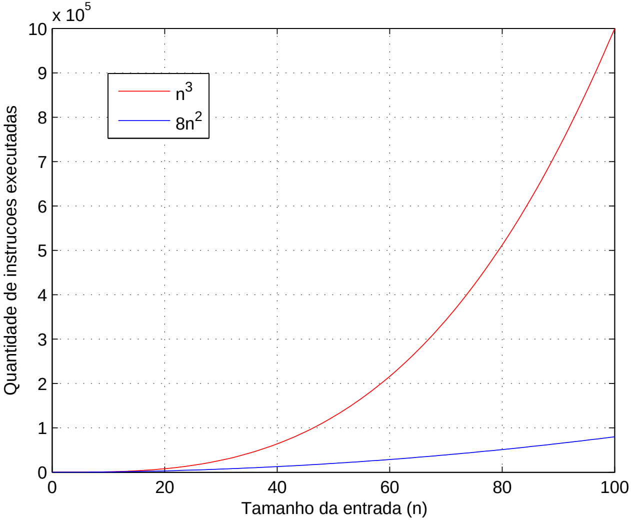
  Os algoritmos têm igual desempenho quando $n = 8$. Já para $n > 8$, o algoritmo A sempre será mais eficiente que B.

Questão 2



1000
900
800
700
600
500
400
300
200
100
0
0
2
4
6
8
10
Quantidade de instrucoes executadas
Tamanho da entrada (n)
n3
8n2

Questão 2



x 10^5
10
9
8
7
6
5
4
3
2
1
0
0
20
40
60
80
100
Quantidade de instrucoes executadas
Tamanho da entrada (n)
n^3
8n^2

Questão 3

- Analise a complexidade no tempo do algoritmo abaixo:

```
MaxMin (V, n)
1. max = V[1]
2. min = V[1]
3. para i = 2 até n faça
4.    se V[i] > max então max = V[i]
5.    se V[i] < min então min = V[i]
```

- Podemos dizer que o algoritmo para e é correto?
- Existe melhor e pior caso?
- O algoritmo é eficiente?

Questão 3

- Sim, o algoritmo para. Isso porque o laço "para" (linha 3) para após n iterações, visto que a variável n é mantida constante durante toda a sua execução. Sim, o algoritmo é correto. No começo de cada iteração do laço "para" (linha 3), as variáveis max e min armazenam o maior e o menor elemento do vetor $A[1 \ldots i-1]$, respectivamente.

- O algoritmo tem complexidade no tempo $O(n)$. É possível representar a complexidade do algoritmo usando a notação assintótica $\Theta(n)$, pois não existe pior ou melhor caso.

- Sim, o algoritmo é eficiente. Isso porque a função que mede sua complexidade no tempo é limitada por um polinômio.

Questão 4

- Analise a complexidade no tempo do algoritmo abaixo:

```
TESTE (A, V, n, w)
para i = 1 até n faça
   para j = 1 até w faça
      V[j] = A[i] + 2
max = V[1]
para j = 2 até w faça
   se max < V[j] então max = V[j]
```

Questão 4

- Analise a complexidade no tempo do algoritmo abaixo.

```
TESTE (A, V, n, w)
para i = 1 até n faça
    para j = 1 até w faça
        V[j] = A[i] + 2
max = V[1]
para j = 2 até w faça
    se max < V[j] então max = V[j]
```

- O algoritmo tem complexidade no tempo $O(nw)$ e não existe pior ou melhor caso.

Questão 6

- Mostre que $7x^2$ é $O(x^3)$.

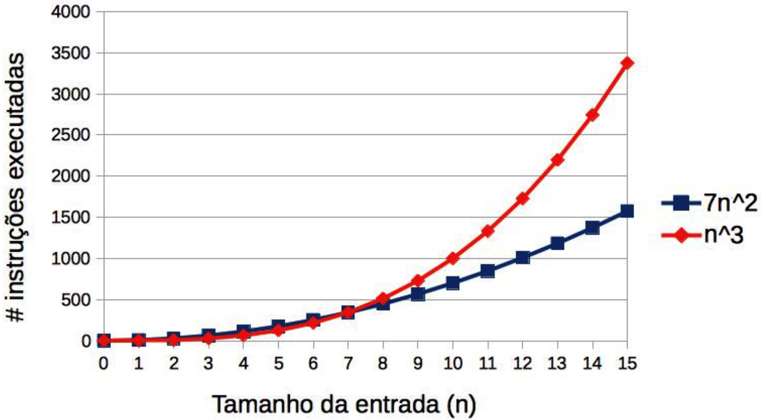
- Solução: Considerando $f(x) = 7x^2$ e $g(x) = x^3$, temos que provar que

$$0 \leq f(x) \leq c\ g(x), \text{ sempre que } x \geq k.$$

- Fixando $c = 1$, tem-se:
  - Para $x = 7$, temos que $f(x) = g(x)$.
  - Para $x > 7$, temos que $f(x) < g(x)$.

- Assim, podemos dizer que $7x^2$ é $O(x^3)$, com $c = 1$ e $k = 7$.

- Só que x^3 não é $O(7x^2)$, logo não são equivalentes.



Questão 7

- Mostre que x^2 é $\Omega(x)$.

- Solução: Considerando $f(x) = x^2$ e $g(x) = x$, temos que provar que

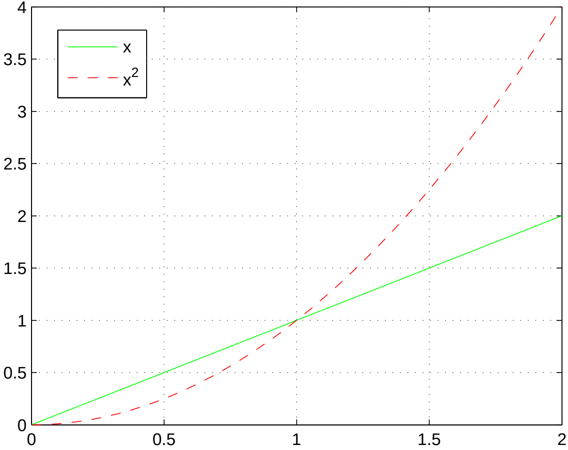
$$0 \leq c\ g(x) \leq f(x), \text{ sempre que } x \geq k.$$

- Fixando $c = 1$, tem-se:
  - Para $x = 1$, temos que $f(x) = g(x)$.
  - Para $x > 1$, temos que $f(x) > g(x)$.

- Assim, podemos dizer que x^2 é $\Omega(x)$, com $c = 1$ e $k = 1$.

Questão 7

- Só que x não é $\Omega(x^2)$, logo não são equivalentes.



Questão 8

- Verifique se as afirmações abaixo são verdadeiras ou falsas:
- $2n^2 + 1000$ é $\Omega(n^2)$.

  A afirmativa é **verdadeira**.

  Para $n \geq 1$ e $c = 1$, tem-se $0 \leq n^2 \leq 2n^2 + 1000$.
- $log(n^2)$ é $\omega(log(n))$.

  A afirmativa é **falsa**.

  Como $log(n^2) = 2log(n)$, as funções são assintoticamente equivalentes, e a notação ω não aceita equivalência.

Questão 8

- 2^{n+1} é $O(2^n)$.

  A afirmativa é **verdadeira**.

  Para $n \geq 1$ e $c = 2$, tem-se $0 \leq 2^{n+1} \leq 2\ 2^n$.

- 2^{2n} é $O(2^n)$.

  A afirmativa é **falsa**. É preciso que $c \geq 2^n$, o que impossível já que c deve ser constante.

Questões 12

- Um algoritmo tradicional tem complexidade $n^{1,5}$, enquanto um algoritmo novo proposto é da ordem de *nlogn*:

  $f(n) = n^{1,5}$
  $g(n) = nlogn$

  Qual algoritmo adotar?

- Uma possível solução:

  $$f(n) = \frac{n^{1,5}}{n} = n^{0,5} \quad \Rightarrow \quad (n^{0,5})^2 = n$$

  $$g(n) = \frac{nlogn}{n} = logn \quad \Rightarrow \quad (logn)^2 = log^2 n$$

- Como n cresce mais rapidamente do que qualquer potência de log, o algoritmo novo é mais eficiente.

Questão 14

a. $O(f+g) = O(f) + O(g)$. **FALSO**.
$O(f+g) = max(O(f), O(g))$.

b. $O(f.g) = O(f).O(g)$. **VERDADEIRO**.

c. $O(k.g) = k.O(g) = O(g)$. **VERDADEIRO**.

d. Se $f = O(g)$, então $g = \Omega(f)$. **VERDADEIRO**.

e. Se $f = O(g)$ e $g = O(h)$, então $f = O(h)$. **VERDADEIRO**.

Questões Subjetivas

05. Resposta correta: D

09. Resposta correta: E

10. Resposta correta: C

11. Resposta correta: B

13. Resposta correta: E

15. Resposta correta: D

16. Resposta correta: C

17. Resposta correta: D

18. Resposta correta: B

19. Resposta correta: A

20. Resposta correta: A

21. Resposta correta: E

22. Resposta correta: A

23. Resposta correta: A