Análise de Algoritmos
Aula 02

Prof. Murilo V. G. da Silva

DINF/UFPR

(material da disciplina: André Guedes, Renato Carmo, Murilo da Silva)

Analisando um algoritmo (sem uso de notação assintótica)

Busca Sequencial:

$\mathrm{S}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
Se $x \geq v[b]$
    Devolva b
Devolva $S(x, v, a, b - 1)$

Analisando um algoritmo (sem uso de notação assintótica)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Analisando um algoritmo (sem uso de notação assintótica)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso “mantra” para a Busca Sequencial:

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso “mantra” para a Busca Sequencial:

(0) Que problema resolve?

Analisando um algoritmo (sem uso de notação assintótica)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?

(1) Está correto?

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso “mantra” para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?
(3) É possível fazer melhor?

Analisando um algoritmo (sem uso de notação assintótica)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?
(3) É possível fazer melhor?

Perguntas (0) e (1):

Analisando um algoritmo (sem uso de notação assintótica)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?
(3) É possível fazer melhor?

Perguntas (0) e (1):

- Resposta para (0): BVO

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?
(3) É possível fazer melhor?

Perguntas (0) e (1):

- Resposta para (0): BVO (*)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?
(3) É possível fazer melhor?

Perguntas (0) e (1):

- Resposta para (0): BVO (*)
- Resposta para (1): Exercício 1

Analisando um algoritmo (sem uso de notação assintótica)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?
(3) É possível fazer melhor?

Perguntas (0) e (1):

- Resposta para (0): BVO (*)
- Resposta para (1): Exercício 1

Perguntas (2) e (3):

Analisando um algoritmo (sem uso de notação assintótica)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?
(3) É possível fazer melhor?

Perguntas (0) e (1):

- Resposta para (0): BVO (*)
- Resposta para (1): Exercício 1

Perguntas (2) e (3):

- Gastaremos boa parte da aula com isso...

Analisando um algoritmo (sem uso de notação assintótica)

Problema de hoje: Busca em Vetor Ordenado (BVO)

Vamos aos nosso "mantra" para a Busca Sequencial:

(0) Que problema resolve?
(1) Está correto?
(2) Quanto custa?
(3) É possível fazer melhor?

Perguntas (0) e (1):

- Resposta para (0): BVO (*)
- Resposta para (1): Exercício 1

Perguntas (2) e (3):

- Gastaremos boa parte da aula com isso...

(*) Observe que Busca sequencial resolve um problema mais geral que BVO, mas nosso foco é BVO aqui, então é correto afirmar que ela resolve BVO.

Quanto Custa?

```
S(x, v, a, b)
Se a > b
    Devolva a − 1
Se x ≥ v[b]
    Devolva b
Devolva S(x, v, a, b − 1)
```

Quanto Custa?

$S(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
Se $x \geq v[b]$
    Devolva b
Devolva $S(x, v, a, b - 1)$

- Qual o significado de “custo”?

Quanto Custa?

$S(x, v, a, b)$
Se $a > b$
Devolva $a - 1$
Se $x \geq v[b]$
Devolva b
Devolva $S(x, v, a, b - 1)$

- Qual o significado de "custo"?

  Vamos estabelecer que, **nesta análise**, "custo" tem o sentido de "tempo máximo de execução em função do tamanho do vetor v".

Quanto Custa?

```
S(x, v, a, b)
1 Se a > b
2     Devolva a − 1
3 Se x ≥ v[b]
4     Devolva b
5 Devolva S(x, v, a, b − 1)
```

- Qual o significado de "custo"?

  Vamos estabelecer que, **nesta análise**, "custo" tem o sentido de "tempo máximo de execução em função do tamanho do vetor v".

- Seja: s_i o tempo para executar a i-ésima linha do Algoritmo S,

Quanto Custa?

```
S(x, v, a, b)
1 Se a > b
2     Devolva a − 1
3 Se x ≥ v[b]
4     Devolva b
5 Devolva S(x, v, a, b − 1)
```

- Qual o significado de "custo"?

  Vamos estabelecer que, **nesta análise**, "custo" tem o sentido de "tempo máximo de execução em função do tamanho do vetor v".

- Seja: s_i o tempo para executar a i-ésima linha do Algoritmo S, $i \in [1..5]$

Quanto Custa?

```
S(x, v, a, b)
1  Se a > b
2      Devolva a − 1
3  Se x ≥ v[b]
4      Devolva b
5  Devolva S(x, v, a, b − 1)
```

- Qual o significado de "custo"?

  Vamos estabelecer que, **<u>nesta análise</u>**, "custo" tem o sentido de "tempo máximo de execução em função do tamanho do vetor v".

- Seja: s_i o tempo para executar a i-ésima linha do Algoritmo S, $i \in [1..5]$

  $T_S(x, v, a, b)$ o tempo para executar S(x,v,a,b)

Quanto Custa?

S(x, v, a, b)
1 Se $a > b$
2 Devolva $a - 1$
3 Se $x \geq v[b]$
4 Devolva b
5 Devolva $S(x, v, a, b - 1)$

- Qual o significado de "custo"?

  Vamos estabelecer que, **nesta análise**, "custo" tem o sentido de "tempo máximo de execução em função do tamanho do vetor v".

- Seja: s_i o tempo para executar a i-ésima linha do Algoritmo S, $i \in [1..5]$

  $T_S(x, v, a, b)$ o tempo para executar S(x,v,a,b)

- Estabelecemos que o "custo" do Algoritmo S é dado pela função:

Quanto Custa?

```
S(x, v, a, b)
1  Se a > b
2      Devolva a − 1
3  Se x ≥ v[b]
4      Devolva b
5  Devolva S(x, v, a, b − 1)
```

- Qual o significado de "custo"?

  Vamos estabelecer que, **<u>nesta análise</u>**, "custo" tem o sentido de "tempo máximo de execução em função do tamanho do vetor v".

- Seja: s_i o tempo para executar a i-ésima linha do Algoritmo S, $i \in [1..5]$

  $T_S(x, v, a, b)$ o tempo para executar S(x,v,a,b)

- Estabelecemos que o "custo" do Algoritmo S é dado pela função:

$$S^+(n) = \max\{T_S(x, v, a, b)\colon\ v \text{ tem } n = b - a + 1 \text{ elementos}\}$$

(3) Quanto custa? (cont.)

```
S(x, v, a, b)
Se a > b
    Devolva a − 1
Se x ≥ v[b]
    Devolva b
Devolva S(x, v, a, b − 1)
```

A função $S^{+}(n)$ aqui representa o *tempo de execução no pior caso*.

(3) Quanto custa? (cont.)

$\mathrm{S}(x, v, a, b)$
Se $a > b$
Devolva $a - 1$
Se $x \geq v[b]$
Devolva b
Devolva $S(x, v, a, b - 1)$

A função $S^{+}(n)$ aqui representa o *tempo de execução no pior caso*.

Poderíamos usar vários outras funções dependendo do que queremos analisar).

(3) Quanto custa? (cont.)

$\mathrm{S}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
Se $x \geq v[b]$
    Devolva b
Devolva $S(x, v, a, b-1)$

A função $S^+(n)$ aqui representa o *tempo de execução no pior caso*.

Poderíamos usar vários outras funções dependendo do que queremos analisar).

$$S^+(n) = \begin{cases} s_1 + s_2, & \text{se } n = 0, \\ s_1 + s_3 + S^+(n-1), & \text{se } n > 0, \end{cases}$$

(3) Quanto custa? (cont.)

$\mathrm{S}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
Se $x \geq v[b]$
    Devolva b
Devolva $S(x, v, a, b - 1)$

A função $S^+(n)$ aqui representa o *tempo de execução no pior caso*.
Poderíamos usar vários outras funções dependendo do que queremos analisar).

$$S^+(n) = \begin{cases} s_1 + s_2, & \text{se } n = 0, \\ s_1 + s_3 + S^+(n-1), & \text{se } n > 0, \end{cases}$$

Resolvendo a recorrência (Exercício 2), temos

(3) Quanto custa? (cont.)

```
S(x, v, a, b)
Se a > b
    Devolva a − 1
Se x ≥ v[b]
    Devolva b
Devolva S(x, v, a, b − 1)
```

A função $S^+(n)$ aqui representa o *tempo de execução no pior caso*.

Poderíamos usar vários outras funções dependendo do que queremos analisar).

$$S^+(n) = \begin{cases} s_1 + s_2, & \text{se } n = 0, \\ s_1 + s_3 + S^+(n-1), & \text{se } n > 0, \end{cases}$$

Resolvendo a recorrência (Exercício 2), temos

$$S^+(n) = c_2 n + c_1,$$

S(x, v, a, b)
Se $a > b$
Devolva $a - 1$
Se $x \geq v[b]$
Devolva b
Devolva $S(x, v, a, b - 1)$

A função $S^+(n)$ aqui representa o *tempo de execução no pior caso*.
Poderíamos usar vários outras funções dependendo do que queremos analisar).

$$S^+(n) = \begin{cases} s_1 + s_2, & \text{se } n = 0, \\ s_1 + s_3 + S^+(n-1), & \text{se } n > 0, \end{cases}$$

Resolvendo a recorrência (Exercício 2), temos

$$S^+(n) = c_2 n + c_1,$$

onde $c_1 = s_1 + s_2$ e $c_2 = s_1 + s_3$ por questão de clareza

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve
  - *Resolver o mesmo problema de interesse (BVO)*

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve
  - *Resolver o mesmo problema de interesse (BVO)*
  - *Estar correto*

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve
  - *Resolver o mesmo problema de interesse (BVO)*
  - *Estar correto*
  - *Custar menos que o algoritmo anterior*

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve
  - *Resolver o mesmo problema de interesse (BVO)*
  - *Estar correto*
  - *Custar menos que o algoritmo anterior*

Importante:

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve
  - *Resolver o mesmo problema de interesse (BVO)*
  - *Estar correto*
  - *Custar menos que o algoritmo anterior*

Importante: A seguir vamos analisar a busca binária. As razões são duas:

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve
  - *Resolver o mesmo problema de interesse (BVO)*
  - *Estar correto*
  - *Custar menos que o algoritmo anterior*

Importante: A seguir vamos analisar a busca binária. As razões são duas:

- Razão imediata: Responder a pertunta (3) do mantra para a *busca sequencial*

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve
  - *Resolver o mesmo problema de interesse (BVO)*
  - *Estar correto*
  - *Custar menos que o algoritmo anterior*

Importante: A seguir vamos analisar a busca binária. As razões são duas:

- Razão imediata: Responder a pertunta (3) do mantra para a *busca sequencial*
- Outra razão: Praticarmos a análise de um outro algoritmo mais sofisticado.

Mantra: (3) É possível fazer melhor?

- Resposta: Sim, usando *Busca Binária*
- Para tal, vamos *Analisar* a Busca Binária e *provar que ela é melhor*

- Note: O Algoritmo de Busca Binária deve
  - *Resolver o mesmo problema de interesse (BVO)*
  - *Estar correto*
  - *Custar menos que o algoritmo anterior*

Importante: A seguir vamos analisar a busca binária. As razões são duas:

- Razão imediata: Responder a pertunta (3) do mantra para a *busca sequencial*
- Outra razão: Praticarmos a análise de um outro algoritmo mais sofisticado.
  (em particular, será um algoritmo para qual a resposta para a pergunta (3) será **não**))

Analisando o Algoritmo de Busca Binária

$\mathrm{B}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

Analisando o Algoritmo de Busca Binária

$\mathrm{B}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m - 1)$
Devolva $B(x, v, m + 1, b)$

Perguntas (0) e (1):

Analisando o Algoritmo de Busca Binária

$\mathrm{B}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

Perguntas (0) e (1):

- Resposta para (0): BVO

Analisando o Algoritmo de Busca Binária

B(x, v, a, b)
Se $a > b$
Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

Perguntas (0) e (1):

- Resposta para (0): BVO
- Resposta para (1): Exercício 3

Analisando o Algoritmo de Busca Binária

B(x, v, a, b)

```
Se a > b
    Devolva a − 1
m ← ⌊(a+b)/2⌋
Se x < v[m]
    Devolva B(x, v, a, m − 1)
Devolva B(x, v, m + 1, b)
```

Perguntas (0) e (1):

- Resposta para (0): BVO
- Resposta para (1): Exercício 3

Perguntas (2) e (3):

Analisando o Algoritmo de Busca Binária

B(x, v, a, b)

```
Se a > b
    Devolva a − 1
m ← ⌊(a+b)/2⌋
Se x < v[m]
    Devolva B(x, v, a, m − 1)
Devolva B(x, v, m + 1, b)
```

Perguntas (0) e (1):

- Resposta para (0): BVO
- Resposta para (1): Exercício 3

Perguntas (2) e (3):

- Gastaremos boa parte da aula com isso...

Analisando o Algoritmo de Busca Binária

$\mathrm{B}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

Perguntas (0) e (1):

- Resposta para (0): BVO
- Resposta para (1): Exercício 3

Perguntas (2) e (3):

- Gastaremos boa parte da aula com isso...
- Em particular, como consequência veremos que o custo da busca binária é menor que o custo da busca sequencial.

Analisando o Algoritmo de Busca Binária

$\mathrm{B}(x, v, a, b)$
Se $a > b$
Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

Perguntas (0) e (1):

- Resposta para (0): BVO
- Resposta para (1): Exercício 3

Perguntas (2) e (3):

- Gastaremos boa parte da aula com isso...
- Em particular, como consequência veremos que o custo da busca binária é menor que o custo da busca sequencial.
  (Respondendo a pergunta (3) da análise da busca sequencial que ficou em aberto)

(2) Quanto custa?

$\mathrm{B}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

(2) Quanto custa?

$\mathrm{B}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m - 1)$
Devolva $B(x, v, m + 1, b)$

Usando o mesmo critério de custo usado para o Algoritmo S.

(2) Quanto custa?

$\mathrm{B}(x, v, a, b)$
1 Se $a > b$
2   Devolva $a - 1$
3 $m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
4 Se $x < v[m]$
5   Devolva $B(x, v, a, m-1)$
6 Devolva $B(x, v, m+1, b)$

Usando o mesmo critério de custo usado para o Algoritmo S.

b_i : tempo para executar a i-ésima linha de B,

(2) Quanto custa?

```
B(x, v, a, b)
1 Se a > b
2     Devolva a − 1
3 m ← ⌊(a+b)/2⌋
4 Se x < v[m]
5     Devolva B(x, v, a, m − 1)
6 Devolva B(x, v, m + 1, b)
```

Usando o mesmo critério de custo usado para o Algoritmo S.

b_i : tempo para executar a i-ésima linha de B,

Com isso a função de custo (*de pior caso*) é:

(2) Quanto custa?

```
B(x, v, a, b)
1 Se a > b
2     Devolva a − 1
3 m ← ⌊(a+b)/2⌋
4 Se x < v[m]
5     Devolva B(x, v, a, m − 1)
6 Devolva B(x, v, m + 1, b)
```

Usando o mesmo critério de custo usado para o Algoritmo S.

b_i : tempo para executar a i-ésima linha de B,

Com isso a função de custo (*de pior caso*) é:

$$B^+(n) = \mathrm{Max}\{B(x, v, a, b) \colon b - a + 1 = n\}.$$

(2) Quanto custa? (cont.)

B(x, v, a, b)

```
Se a > b
    Devolva a − 1
m ← ⌊(a+b)/2⌋
Se x < v[m]
    Devolva B(x, v, a, m − 1)
Devolva B(x, v, m + 1, b)
```

(2) Quanto custa? (cont.)

B(x, v, a, b)

```
Se a > b
      Devolva a − 1
m ← ⌊(a+b)/2⌋
Se x < v[m]
      Devolva B(x, v, a, m − 1)
Devolva B(x, v, m + 1, b)
```

$$B^+(n) = \begin{cases} c_1', & \text{se } n = 0, \\ c_2' + \text{Max}\{B^+((m+1) - a + 1), B^+(b - (m+1) + 1)\}, & \text{se } n > 0, \end{cases}$$

(2) Quanto custa? (cont.)

B(x, v, a, b)

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \lfloor \frac{a+b}{2} \rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

$$B^+(n) = \begin{cases} c_1', & \text{se } n = 0, \\ c_2' + \text{Max}\{B^+((m+1) - a + 1), B^+(b - (m+1) + 1)\}, & \text{se } n > 0, \end{cases}$$

$$\begin{aligned} c_1' &= b_1 + b_2, \\ c_2' &= b_1 + b_3 + b_4, \\ m &= \lfloor \frac{a+b}{2} \rfloor, \end{aligned}$$

(2) Quanto custa? (cont.)

B(x, v, a, b)

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m - 1)$
Devolva $B(x, v, m + 1, b)$

(2) Quanto custa? (cont.)

B(x, v, a, b)

```
Se a > b
    Devolva a − 1
m ← ⌊(a+b)/2⌋
Se x < v[m]
    Devolva B(x, v, a, m − 1)
Devolva B(x, v, m + 1, b)
```

A equação para B^+ pode ser simplificada (Exercício 4) para

(2) Quanto custa? (cont.)

$\mathrm{B}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

A equação para B^+ pode ser simplificada (Exercício 4) para

$$B^+(n) = \begin{cases} c_1', & \text{se } n = 0, \\ c_2' + B^+\{\lfloor \frac{n-1}{2} \rfloor\}, & \text{se } n > 0, \end{cases}$$

(2) Quanto custa? (cont.)

$\mathrm{B}(x, v, a, b)$

Se $a > b$
    Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
    Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

A equação para B^+ pode ser simplificada (Exercício 4) para

$$B^+(n) = \begin{cases} c'_1, & \text{se } n = 0, \\ c'_2 + B^+\{\lfloor \frac{n-1}{2} \rfloor\}, & \text{se } n > 0, \end{cases}$$

Pelo Exercício 5 chegamos a:

(2) Quanto custa? (cont.)

$\mathrm{B}(x, v, a, b)$
Se $a > b$
Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

A equação para B^+ pode ser simplificada (Exercício 4) para

$$B^+(n) = \begin{cases} c_1', & \text{se } n = 0, \\ c_2' + B^+\{\lfloor \frac{n-1}{2} \rfloor\}, & \text{se } n > 0, \end{cases}$$

Pelo Exercício 5 chegamos a:

$$B^+(n) \leq c_1' + c_2' \lg n, \forall n > 0.$$

(2) Quanto custa? (cont.)

$\mathrm{B}(x, v, a, b)$
Se $a > b$
Devolva $a - 1$
$m \leftarrow \left\lfloor \frac{a+b}{2} \right\rfloor$
Se $x < v[m]$
Devolva $B(x, v, a, m-1)$
Devolva $B(x, v, m+1, b)$

A equação para B^+ pode ser simplificada (Exercício 4) para

$$B^+(n) = \begin{cases} c_1', & \text{se } n = 0, \\ c_2' + B^+\{\lfloor \frac{n-1}{2} \rfloor\}, & \text{se } n > 0, \end{cases}$$

Pelo Exercício 5 chegamos a:

$$B^+(n) \leq c_1' + c_2' \lg n, \forall n > 0.$$

- Antes de responder (3) para B, vamos comparar o custo de B com S
  (assim responderemos a pergunta (3) para o algoritmo S (busca sequencial))

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1}$$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1' / \lg n}{c_2 + c_1/n}\right)$$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1'/\lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right)$$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1'/\lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1'/\lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

onde

$$c_3' = \frac{c_2' + c_1'}{c_2}.$$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1' / \lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

onde

$$c_3' = \frac{c_2' + c_1'}{c_2}.$$

Veja que:

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1'/\lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

onde

$$c_3' = \frac{c_2' + c_1'}{c_2}.$$

Veja que: $0 \leq \lim \frac{B^+(n)}{S^+(n)}$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1'/\lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

onde

$$c_3' = \frac{c_2' + c_1'}{c_2}.$$

Veja que: $0 \leq \lim \frac{B^+(n)}{S^+(n)} \leq \lim c_3' \frac{\lg n}{n}$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1'/\lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

onde

$$c_3' = \frac{c_2' + c_1'}{c_2}.$$

Veja que: $0 \leq \lim \frac{B^+(n)}{S^+(n)} \leq \lim c_3' \frac{\lg n}{n} = 0.$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1' / \lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

onde

$$c_3' = \frac{c_2' + c_1'}{c_2}.$$

Veja que: $0 \leq \lim \frac{B^+(n)}{S^+(n)} \leq \lim c_3' \frac{\lg n}{n} = 0$. Portanto:

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1'/\lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

onde

$$c_3' = \frac{c_2' + c_1'}{c_2}.$$

Veja que: $0 \leq \lim \frac{B^+(n)}{S^+(n)} \leq \lim c_3' \frac{\lg n}{n} = 0$. Portanto:

$$\lim \frac{B^+(n)}{S^+(n)} = 0,$$

Comparando Busca Sequencial e Binária

A solução para o BVO dada pelo Algoritmo B será melhor que a dada pelo Algoritmo S nas instâncias (x, v, a, b) onde $n = b - a + 1$ se satisfizer:

$$B^+(n) < S^+(n),$$

ou, equivalentemente,

$$\frac{B^+(n)}{S^+(n)} < 1. \tag{1}$$

Como para todo $n \geq 2$,

$$\frac{B^+(n)}{S^+(n)} \leq \frac{c_2' \lg n + c_1'}{c_2 n + c_1} = \frac{\lg n}{n}\left(\frac{c_2' + c_1'/\lg n}{c_2 + c_1/n}\right) \leq \frac{\lg n}{n}\left(\frac{c_2' + c_1'}{c_2}\right) = c_3' \frac{\lg n}{n},$$

onde

$$c_3' = \frac{c_2' + c_1'}{c_2}.$$

Veja que: $0 \leq \lim \frac{B^+(n)}{S^+(n)} \leq \lim c_3' \frac{\lg n}{n} = 0$. Portanto:

$$\lim \frac{B^+(n)}{S^+(n)} = 0,$$

independentemente do valor de c_3'.

Comparando Busca Sequencial e Binária

Teorema 1

Para quaisquer valores de $s_1, \ldots, s_5, b_1, \ldots, b_6 > 0$ temos $\lim \frac{B^+(n)}{S^+(n)} = 0$.

Comparando Busca Sequencial e Binária

Teorema 1

Para quaisquer valores de $s_1, \ldots, s_5, b_1, \ldots, b_6 > 0$ temos $\lim \frac{B^+(n)}{S^+(n)} = 0$.

Corolário 2 (T.1)

Para quaisquer valores de $s_1, \ldots, s_5, b_1, \ldots, b_6, k > 0$, existe $n_k \in \mathbb{N}$, tal que

$$B^+(n) < \frac{S^+(n)}{k}, \forall n \geq n_k.$$

Comparando Busca Sequencial e Binária

Corolário 3 (T.1)

Independentemente dos dispositivos que executam os respectivos algoritmos, para todo $k > 0$ existe $n_k \in \mathbb{N}$ tal que $B^+(n) < \frac{S^+(n)}{k}, \forall n \geq n_k$.

Comparando Busca Sequencial e Binária

Corolário 3 (T.1)

Independentemente dos dispositivos que executam os respectivos algoritmos, para todo $k > 0$ existe $n_k \in \mathbb{N}$ tal que $B^+(n) < \frac{S^+(n)}{k}, \forall n \geq n_k$.

Corolário 3 em outras palavras

Para todo $k > 0$, existe $n_k \in \mathbb{N}$ tal que $\forall n \geq n_k$,

existem instâncias (x_S, v_S, b, a) e (x_B, v_B, b, a) do BVO tais que $b - a + 1 = n$ e a execução de $B(x_B, v_B, b, a)$ é k vezes mais rápido que a de $S(x_S, v_S, b, a)$,

quaisquer que sejam os dispositivos que os executam.

Comparando Busca Sequencial e Binária

Corolário 3 (T.1)

Independentemente dos dispositivos que executam os respectivos algoritmos, para todo $k > 0$ existe $n_k \in \mathbb{N}$ tal que $B^+(n) < \frac{S^+(n)}{k}, \forall n \geq n_k$.

Corolário 3 em outras palavras

Para todo $k > 0$, existe $n_k \in \mathbb{N}$ tal que $\forall n \geq n_k$,

existem instâncias (x_S, v_S, b, a) e (x_B, v_B, b, a) do BVO tais que $b - a + 1 = n$ e a execução de $B(x_B, v_B, b, a)$ é k vezes mais rápido que a de $S(x_S, v_S, b, a)$,

quaisquer que sejam os dispositivos que os executam.

Corolário 4 (T.1)

O Algoritmo B é mais eficiente que o Algoritmo S para o BVO na *análise de pior caso*.

Comparando Busca Sequencial e Binária

Corolário 3 (T.1)

Independentemente dos dispositivos que executam os respectivos algoritmos, para todo $k > 0$ existe $n_k \in \mathbb{N}$ tal que $B^+(n) < \frac{S^+(n)}{k}, \forall n \geq n_k$.

Corolário 3 em outras palavras

Para todo $k > 0$, existe $n_k \in \mathbb{N}$ tal que $\forall n \geq n_k$,

existem instâncias (x_S, v_S, b, a) e (x_B, v_B, b, a) do BVO tais que $b - a + 1 = n$ e a execução de $B(x_B, v_B, b, a)$ é k vezes mais rápido que a de $S(x_S, v_S, b, a)$,

quaisquer que sejam os dispositivos que os executam.

Corolário 4 (T.1)

O Algoritmo B é mais eficiente que o Algoritmo S para o BVO na *análise de pior caso*.

Obs: Finalmente respondemos a pergunta (3) do mantra para busca sequencial!

Mantra: (3) É possível fazer melhor? (Busca Binária)

Definição de instâncias de tamanho n

$I(n) = \{\text{instâncias } \mathcal{I} = (x, v, a, b) \text{ de BVO } \; ; \, b - a + 1 = n\}$

- Seja A um algoritmo determinístico para BVO. (supostamente melhor que B)

Mantra: (3) É possível fazer melhor? (Busca Binária)

Definição de instâncias de tamanho n

$I(n) = \{\text{instâncias } \mathcal{I} = (x, v, a, b) \text{ de BVO } \ ; b - a + 1 = n\}$

- Seja A um algoritmo determinístico para BVO. (supostamente melhor que B)
- Dado $n \geq 0$ seja $T_A(n)$ a árvore binária com raíz $i - a$, onde i é o índice do primeiro elemento de v comparado com x na execução de A algum $\mathcal{I} \in I(n)$.

Mantra: (3) É possível fazer melhor? (Busca Binária)

Definição de instâncias de tamanho n

$I(n) = \{\text{instâncias } \mathcal{I} = (x, v, a, b) \text{ de BVO } \; ; \; b - a + 1 = n\}$

- Seja A um algoritmo determinístico para BVO. (supostamente melhor que B)
- Dado $n \geq 0$ seja $T_A(n)$ a árvore binária com raíz $i - a$, onde i é o índice do primeiro elemento de v comparado com x na execução de A algum $\mathcal{I} \in I(n)$.
  (dica para quem está confuso: pense no caso particular que estamos considerando o vetor todo, ou seja $a = 0$ e $b = n - 1$)

Mantra: (3) É possível fazer melhor? (Busca Binária)

Definição de instâncias de tamanho n

$I(n) = \{\text{instâncias } \mathcal{I} = (x, v, a, b) \text{ de BVO } \; ; \; b - a + 1 = n\}$

- Seja A um algoritmo determinístico para BVO. (supostamente melhor que B)
- Dado $n \geq 0$ seja $T_A(n)$ a árvore binária com raíz $i - a$, onde i é o índice do primeiro elemento de v comparado com x na execução de A algum $\mathcal{I} \in I(n)$.
  (dica para quem está confuso: pense no caso particular que estamos considerando o vetor todo, ou seja $a = 0$ e $b = n - 1$)
- A raíz de $L(T_A(n))$, a subárvore esquerda de $T_A(n)$, tem $j - a$ onde j é o índice do segundo elemento de v comparado com x na execução de A caso $x < v[i]$.

Mantra: (3) É possível fazer melhor? (Busca Binária)

Definição de instâncias de tamanho n

$I(n) = \{\text{instâncias } \mathcal{I} = (x, v, a, b) \text{ de BVO } ;\ b - a + 1 = n\}$

- Seja A um algoritmo determinístico para BVO. (supostamente melhor que B)
- Dado $n \geq 0$ seja $T_A(n)$ a árvore binária com raíz $i - a$, onde i é o índice do primeiro elemento de v comparado com x na execução de A algum $\mathcal{I} \in I(n)$.
  (dica para quem está confuso: pense no caso particular que estamos considerando o vetor todo, ou seja $a = 0$ e $b = n - 1$)
- A raíz de $L(T_A(n))$, a subárvore esquerda de $T_A(n)$, tem $j - a$ onde j é o índice do segundo elemento de v comparado com x na execução de A caso $x < v[i]$.
- A raíz de $R(T_A(n))$, a subárvore direita de $T_A(n)$, tem $k - a$ onde k é o índice do segundo elemento de v comparado com x na execução de A caso $x \geq v[i]$.

Definição de instâncias de tamanho n

$I(n) = \{\text{instâncias } \mathcal{I} = (x, v, a, b) \text{ de BVO } \; ; b - a + 1 = n\}$

- Seja A um algoritmo determinístico para BVO. (supostamente melhor que B)
- Dado $n \geq 0$ seja $T_A(n)$ a árvore binária com raíz $i - a$, onde i é o índice do primeiro elemento de v comparado com x na execução de A algum $\mathcal{I} \in I(n)$.
  (dica para quem está confuso: pense no caso particular que estamos considerando o vetor todo, ou seja $a = 0$ e $b = n - 1$)
- A raíz de $L(T_A(n))$, a subárvore esquerda de $T_A(n)$, tem $j - a$ onde j é o índice do segundo elemento de v comparado com x na execução de A caso $x < v[i]$.
- A raíz de $R(T_A(n))$, a subárvore direita de $T_A(n)$, tem $k - a$ onde k é o índice do segundo elemento de v comparado com x na execução de A caso $x \geq v[i]$.
- E assim por diante ...

Mantra: (3) É possível fazer melhor? (Busca Binária)

- A árvore $T_A(n)$ é uma representação de todas as sequências de comparações possíveis numa execução do algoritmo A sobre uma instância de $I(n)$.

Mantra: (3) É possível fazer melhor? (Busca Binária)

- A árvore $T_A(n)$ é uma representação de todas as sequências de comparações possíveis numa execução do algoritmo A sobre uma instância de $I(n)$.

- Cada folha de $T_A(n)$ representa o fim da execução de A com uma das possíveis respostas.

Mantra: (3) É possível fazer melhor? (Busca Binária)

- A árvore $T_A(n)$ é uma representação de todas as sequências de comparações possíveis numa execução do algoritmo A sobre uma instância de $I(n)$.
- Cada folha de $T_A(n)$ representa o fim da execução de A com uma das possíveis respostas. (note: o tamanho do ramo até a folha é o número de comparações que A fez)

Mantra: (3) É possível fazer melhor? (Busca Binária)

- A árvore $T_A(n)$ é uma representação de todas as sequências de comparações possíveis numa execução do algoritmo A sobre uma instância de $I(n)$.
- Cada folha de $T_A(n)$ representa o fim da execução de A com uma das possíveis respostas. (note: o tamanho do ramo até a folha é o número de comparações que A fez)
- A árvore $T_A(n)$ precisa ter ao menos $n + 1$ folhas pois o conjunto das instâncias em $I(n)$ admite $n + 1$ respostas distintas.

Mantra: (3) É possível fazer melhor? (Busca Binária)

- A árvore $T_A(n)$ é uma representação de todas as sequências de comparações possíveis numa execução do algoritmo A sobre uma instância de $I(n)$.
- Cada folha de $T_A(n)$ representa o fim da execução de A com uma das possíveis respostas. (note: o tamanho do ramo até a folha é o número de comparações que A fez)
- A árvore $T_A(n)$ precisa ter ao menos $n+1$ folhas pois o conjunto das instâncias em $I(n)$ admite $n+1$ respostas distintas.
- Uma árvore binária com $n+1$ folhas tem altura pelo menos $\lg(n+1)$ (Exerc. 6).

Mantra: (3) É possível fazer melhor? (Busca Binária)

- A árvore $T_A(n)$ é uma representação de todas as sequências de comparações possíveis numa execução do algoritmo A sobre uma instância de $I(n)$.
- Cada folha de $T_A(n)$ representa o fim da execução de A com uma das possíveis respostas. (note: o tamanho do ramo até a folha é o número de comparações que A fez)
- A árvore $T_A(n)$ precisa ter ao menos $n+1$ folhas pois o conjunto das instâncias em $I(n)$ admite $n+1$ respostas distintas.
- Uma árvore binária com $n+1$ folhas tem altura pelo menos $\lg(n+1)$ (Exerc. 6).
- Portanto, existe uma instância $\mathcal{I} = (x, v, a, b) \in I(n)$ para a qual a execução de $A(\mathcal{I})$ faz pelo menos $\lg(n+1)$ comparações.

Mantra: (3) É possível fazer melhor? (Busca Binária)

- A árvore $T_A(n)$ é uma representação de todas as sequências de comparações possíveis numa execução do algoritmo A sobre uma instância de $I(n)$.
- Cada folha de $T_A(n)$ representa o fim da execução de A com uma das possíveis respostas. (note: o tamanho do ramo até a folha é o número de comparações que A fez)
- A árvore $T_A(n)$ precisa ter ao menos $n+1$ folhas pois o conjunto das instâncias em $I(n)$ admite $n+1$ respostas distintas.
- Uma árvore binária com $n+1$ folhas tem altura pelo menos $\lg(n+1)$ (Exerc. 6).
- Portanto, existe uma instância $\mathcal{I} = (x, v, a, b) \in I(n)$ para a qual a execução de $A(\mathcal{I})$ faz pelo menos $\lg(n+1)$ comparações.
- Com isso: o tempo de execução de $A(\mathcal{I})$ é de pelo menos $\lg(n+1)$ unidades.

Mantra: (3) É possível fazer melhor? (Busca Binária)

Teorema 5

Para todo algoritmo determinístico A para o BVO existe uma instância (x, v, a, b) do problema tal que a execução de $A(x, v, a, b)$ faz pelo menos $\lg(b - a + 2)$ comparações de x com elementos de v.

Mantra: (3) É possível fazer melhor? (Busca Binária)

Teorema 5

Para todo algoritmo determinístico A para o BVO existe uma instância (x, v, a, b) do problema tal que a execução de $A(x, v, a, b)$ faz pelo menos $\lg(b - a + 2)$ comparações de x com elementos de v.

Prova: Não vamos fazer aqui, mas estudantes que gostam de desafios podem fazer.

Mantra: (3) É possível fazer melhor? (Busca Binária)

Teorema 5

Para todo algoritmo determinístico A para o BVO existe uma instância (x, v, a, b) do problema tal que a execução de $A(x, v, a, b)$ faz pelo menos $\lg(b - a + 2)$ comparações de x com elementos de v.

Prova: Não vamos fazer aqui, mas estudantes que gostam de desafios podem fazer.

(dica: basta formalizar os pontos colocados nos dois slides anteriores)

Mantra: (3) É possível fazer melhor? (Busca Binária)

Teorema 5

Para todo algoritmo determinístico A para o BVO existe uma instância (x, v, a, b) do problema tal que a execução de $A(x, v, a, b)$ faz pelo menos $\lg(b - a + 2)$ comparações de x com elementos de v.

Prova: Não vamos fazer aqui, mas estudantes que gostam de desafios podem fazer.
(dica: basta formalizar os pontos colocados nos dois slides anteriores)

- Moral da história: Não é possível fazer melhor do que o Algoritmo B no mesmo sentido em que o Algoritmo B é melhor que o Algoritmo S.

Mantra: (3) É possível fazer melhor? (Busca Binária)

Teorema 5

Para todo algoritmo determinístico A para o BVO existe uma instância (x, v, a, b) do problema tal que a execução de $A(x, v, a, b)$ faz pelo menos $\lg(b - a + 2)$ comparações de x com elementos de v.

Prova: Não vamos fazer aqui, mas estudantes que gostam de desafios podem fazer.
(dica: basta formalizar os pontos colocados nos dois slides anteriores)

- Moral da história: Não é possível fazer melhor do que o Algoritmo B no mesmo sentido em que o Algoritmo B é melhor que o Algoritmo S.

Mais formalmente:

Mantra: (3) É possível fazer melhor? (Busca Binária)

Teorema 5

Para todo algoritmo determinístico A para o BVO existe uma instância (x, v, a, b) do problema tal que a execução de $A(x, v, a, b)$ faz pelo menos $\lg(b - a + 2)$ comparações de x com elementos de v.

Prova: Não vamos fazer aqui, mas estudantes que gostam de desafios podem fazer. (dica: basta formalizar os pontos colocados nos dois slides anteriores)

- Moral da história: Não é possível fazer melhor do que o Algoritmo B no mesmo sentido em que o Algoritmo B é melhor que o Algoritmo S.

Mais formalmente:

Corolário 6 (T.5)

Não existe algoritmo A para o BVO satisfazendo

$$\lim \frac{A^{+}(n)}{B^{+}(n)} = 0,$$

onde $A^{+}(n) = \text{Max}\{T_A(x, v, a, b) \colon b - a + 1 = n\}$ e $T_A(x, v, a, b)$ é o tempo de execução de $A(x, v, a, b)$.