"É o hardware que torna um computador rápido. É o software que transforma um computador rápido em lento" (Craig Bruce).

Busca Binária

Paulo Ricardo Lisboa de Almeida

Alguns conteúdos e exemplos foram baseados nas aulas do Professor Renato Carmo - www.inf.ufpr.br/renato

UNIVERSIDADE FEDERAL DO PARANÁ

Vetor ordenado

Um vetor v indexado por $[a..b]$ é ordenado se seus elementos estão em **ordem não decrescente.** Ou seja, $v[i] \leq v[j], \forall i < j$.

Criar um algoritmo que dado um valor *x*, busca a posição onde x deve ocupar no vetor de forma a manter o vetor ordenado.

Da aula passada

Usamos um algoritmo sequencial ingênuo para resolver o problema

Busca Binária

Podemos fazer **muito** melhor

Considere o exemplo, onde desejamos encontrar a posição onde inserir o número *x = 8*

Podemos “partir o vetor ao meio”

i	1	2	3	4	5	6	7	8
v[i]	4	8	8	15	16	16	23	42

Busca Binária

Podemos fazer **muito** melhor

Considere o exemplo, onde desejamos encontrar a posição onde inserir o número *x = 8*

Podemos “partir o vetor ao meio”

Se x é menor que o elemento central, podemos descartar tudo que está a direita. Caso contrário, podemos descartar tudo que está a esquerda

i	1	2	3	4	5	6	7	8
v[i]	4	8	8	15	16	16	23	42

Busca Binária

Podemos fazer **muito** melhor

Considere o exemplo, onde desejamos encontrar a posição onde inserir o número *x = 8*

Podemos “partir o vetor ao meio”

Se x é menor que o elemento central, podemos descartar tudo que está a direita. Caso contrário, podemos descartar tudo que está a esquerda

O processo se repete

Sempre descartando cerca de 50% do vetor

i	1	2	3
v[i]	4	8	8

Busca Binária

Podemos fazer **muito** melhor

Considere o exemplo, onde desejamos encontrar a posição onde inserir o número *x = 8*

Podemos “partir o vetor ao meio”

Se x é menor que o elemento central, podemos descartar tudo que está a direita. Caso contrário, podemos descartar tudo que está a esquerda

O processo se repete

Sempre descartando cerca de 50% do vetor

i	3	
v[i]	8	

Busca Binária

```
função buscaBinaria (x,v,a,b)
entrada: vetor v ordenado, indexado por [a..b], com a ≤ b, e um valor x a ser buscado
saída: o menor m ∈ [a-1..b], tal que x < v[i] ∀i ∈ [m+1..b]
se a > b
     retorne a-1

m ← ⌊(a + b)/2⌋

se x < v[m]
     retorne buscaBinaria (x,v,a,m-1)
retorne buscaBinaria (x,v,m+1,b)
```

$\lfloor x \rfloor$ representa o piso de x. O piso é o maior inteiro menor ou igual a x, ou seja $\lfloor x \rfloor = max\{z \in \mathbb{Z} | \ \leq x\}$.

Faça Você Mesmo

Faça o teste de mesa considerando o seguinte vetor e instâncias do problema

a. BuscaBinária(8, v, 1, 8)
b. BuscaBinária(20, v, 1, 8)

i	1	2	3	4	5	6	7	8
v[i]	4	8	8	15	16	16	23	42

```
função buscaBinaria (x,v,a,b)
entrada: vetor v ordenado, indexado por [a..b], com a ≤ b, e um valor x a ser buscado
saída: o menor m ∈ [a-1..b], tal que x < v[i] ∀i ∈ [m+1..b]
se a > b
     retorne a-1

m ← ⌊(a + b)/2⌋

se x < v[m]
     retorne buscaBinaria (x,v,a,m-1)
retorne buscaBinaria (x,v,m+1,b)
```

Busca Binária

$$C(x,v,a,b) = \begin{cases} 0, & \text{se } a > b, \\ 1 + C(x,v,a,m-1), & \text{se } x < v[m], \\ 1 + C(x,v,m+1,b), & \text{se } x \geq v[m] \end{cases}$$

```
função buscaBinaria (x,v,a,b)
entrada: vetor v ordenado, indexado por [a..b], com a ≤ b, e um valor x a ser buscado
saída: o menor m ∈ [a-1..b], tal que x < v[i] ∀i ∈ [m+1..b]
se a > b
      retorne a-1
m ← ⌊(a + b)/2⌋
se x < v[m]
      retorne buscaBinaria (x,v,a,m-1)
retorne buscaBinaria (x,v,m+1,b)
```

Pior Caso

$$C^+(n) = max\{C(x, v, a, b) | |(x, v, a, b)| = n\}$$

$$C^+(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + max\{C^+(|(x, v, a, m-1)|), C^+(|(x, v, m+1, b)|)\}, & \text{se } n > 0 \end{cases}$$

Tamanho do vetor

O tamanho dos vetores é dado por

$$C^+(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + max\{C^+(\boxed{|(x, v, a, m-1)|}), C^+(\boxed{|(x, v, m+1, b)|})\}, & \text{se } n > 0 \end{cases}$$

$$|(a, m-1)| = \left\lfloor \frac{|(a,b)| - 1}{2} \right\rfloor = \left\lfloor \frac{n-1}{2} \right\rfloor$$

$$|(m+1, b)| = \left\lfloor \frac{|(a,b)|}{2} \right\rfloor = \left\lfloor \frac{n}{2} \right\rfloor$$

Tamanho do vetor - Prova

$|(a, m-1)| = \left\lfloor \frac{|(a,b)| - 1}{2} \right\rfloor$ Lembrando que $m = \left\lfloor \frac{a+b}{2} \right\rfloor$ e $|(a,b)| = b - a + 1$

Tamanho do vetor - Prova

$|(a, m-1)| = \left\lfloor \frac{|(a,b)| - 1}{2} \right\rfloor$ Lembrando que $m = \left\lfloor \frac{a+b}{2} \right\rfloor$ e $|(a,b)| = b - a + 1$

$m - 1 - a + 1 = \left\lfloor \frac{b - a + 1 - 1}{2} \right\rfloor$

Tamanho do vetor - Prova

$$|(a, m-1)| = \left\lfloor \frac{|(a,b)| - 1}{2} \right\rfloor$$ Lembrando que $$m = \left\lfloor \frac{a+b}{2} \right\rfloor$$ e $$|(a,b)| = b - a + 1$$

$$m - 1 - a + 1 = \left\lfloor \frac{b - a + 1 - 1}{2} \right\rfloor$$

$$m - a = \left\lfloor \frac{b - a + 1 - 1}{2} \right\rfloor$$

Tamanho do vetor - Prova

$|(a, m-1)| = \left\lfloor \frac{|(a,b)| - 1}{2} \right\rfloor$ Lembrando que $m = \left\lfloor \frac{a+b}{2} \right\rfloor$ e $|(a,b)| = b - a + 1$

$$m - 1 - a + 1 = \left\lfloor \frac{b - a + 1 - 1}{2} \right\rfloor$$

$$m - a = \left\lfloor \frac{b - a + 1 - 1}{2} \right\rfloor$$

$$\left\lfloor \frac{a+b}{2} \right\rfloor - a = \left\lfloor \frac{b - a}{2} \right\rfloor$$

Tamanho do vetor - Prova

$$|(a, m-1)| = \left\lfloor \frac{|(a,b)|-1}{2} \right\rfloor$$

Lembrando que $m = \left\lfloor \frac{a+b}{2} \right\rfloor$ e $|(a,b)| = b - a + 1$

$$m - 1 - a + 1 = \left\lfloor \frac{b-a+1-1}{2} \right\rfloor$$

$$m - a = \left\lfloor \frac{b-a+1-1}{2} \right\rfloor$$

$$\left\lfloor \frac{a+b}{2} \right\rfloor - a = \left\lfloor \frac{b-a}{2} \right\rfloor$$

Caso 1: Se a+b é par

$$\frac{a+b}{2} - a = \frac{b-a}{2}$$

$$\frac{a+b}{2} - \frac{2a}{2} = \frac{b-a}{2}$$

$$\frac{b-a}{2} = \frac{b-a}{2}$$

Tamanho do vetor - Prova

$$|(a, m-1)| = \left\lfloor \frac{|(a,b)|-1}{2} \right\rfloor$$

Lembrando que $m = \left\lfloor \frac{a+b}{2} \right\rfloor$ e $|(a,b)| = b - a + 1$

$$m - 1 - a + 1 = \left\lfloor \frac{b-a+1-1}{2} \right\rfloor$$

$$m - a = \left\lfloor \frac{b-a+1-1}{2} \right\rfloor$$

$$\left\lfloor \frac{a+b}{2} \right\rfloor - a = \left\lfloor \frac{b-a}{2} \right\rfloor$$

Caso 1: Se a+b é par

$$\frac{a+b}{2} - a = \frac{b-a}{2}$$

$$\frac{a+b}{2} - \frac{2a}{2} = \frac{b-a}{2}$$

$$\frac{b-a}{2} = \frac{b-a}{2}$$

Caso 2: Se a+b é ímpar

$$\frac{a+b-1}{2} - a = \frac{b-a-1}{2}$$

$$\frac{a+b-1}{2} - \frac{2a}{2} = \frac{b-a-1}{2}$$

$$\frac{b-a-1}{2} = \frac{b-a-1}{2}$$

Tamanho do vetor - Prova

$$|(m+1,b)| = \left\lfloor \frac{|(a,b)|}{2} \right\rfloor$$

Lembrando que $m = \left\lfloor \frac{a+b}{2} \right\rfloor$ e $|(a,b)| = b - a + 1$

Tamanho do vetor - Prova

$$|(m+1,b)| = \left\lfloor \frac{|(a,b)|}{2} \right\rfloor$$

$$b - m - 1 + 1 = \left\lfloor \frac{b-a+1}{2} \right\rfloor$$

$$b - \left\lfloor \frac{a+b}{2} \right\rfloor = \left\lfloor \frac{b-a+1}{2} \right\rfloor$$

Caso 1: Se a+b é par

$$b - \frac{a+b}{2} = \frac{b-a+1-1}{2}$$

$$\frac{2b}{2} - \frac{a+b}{2} = \frac{b-a}{2}$$

$$\frac{b-a}{2} = \frac{b-a}{2}$$

Caso 2: Se a+b é ímpar

$$b - \frac{a+b-1}{2} = \frac{b-a+1}{2}$$

$$\frac{2b}{2} - \frac{a+b-1}{2} = \frac{b-a+1}{2}$$

$$\frac{b-a+1}{2} = \frac{b-a+1}{2}$$

Pior Caso

Substituindo

$$C^+(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + max\{C^+(\left\lfloor \frac{n-1}{2} \right\rfloor), C^+(\left\lfloor \frac{n}{2} \right\rfloor)\}, & \text{se } n > 0 \end{cases}$$

Pior Caso

Substituindo

$$C^+(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + max\{C^+(\left\lfloor \frac{n-1}{2} \right\rfloor), C^+(\left\lfloor \frac{n}{2} \right\rfloor)\}, & \text{se } n > 0 \end{cases}$$

$$C^+(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + C^+(\left\lfloor \frac{n}{2} \right\rfloor), & \text{se } n > 0 \end{cases}$$

Obs.: ainda temos uma complicação se considerarmos que *n* pode ser par ou ímpar. Mas vamos ignorar isso (precisaríamos de alguns passos extras para nos convencer de que o resultado é o mesmo).

Recorrência

$$C^+(n) = 1 + C^+(\left\lfloor \frac{n}{2} \right\rfloor) = \dots$$

Recorrência

$$C^+(n) = 1 + C^+(\left\lfloor \frac{n}{2} \right\rfloor) = 1 + 1 + C^+(\left\lfloor \frac{\lfloor (n/2) \rfloor}{2} \right\rfloor)$$

$$= 1 + 1 + 1 + C^+(\left\lfloor \frac{\lfloor (\lfloor [n/2] \rfloor/2) \rfloor}{2} \right\rfloor) = \dots$$

Recorrência

$$C^{+}(n) = 1 + C^{+}(\left\lfloor \frac{n}{2} \right\rfloor) = 1 + 1 + C^{+}(\left\lfloor \frac{\lfloor (n/2) \rfloor}{2} \right\rfloor)$$

$$= 1 + 1 + 1 + C^{+}(\left\lfloor \frac{\lfloor (\lfloor [n/2] \rfloor /2) \rfloor}{2} \right\rfloor) = \dots$$

Teorema

$$\left\lfloor \frac{\left\lfloor \frac{n}{2^k} \right\rfloor}{2} \right\rfloor = \left\lfloor \frac{n}{2^{k+1}} \right\rfloor$$

$$, \forall n, k \in \mathbb{N}$$

Recorrência

$$C^+(n) = 1 + C^+(\left\lfloor \frac{n}{2} \right\rfloor) = 1 + 1 + C^+(\left\lfloor \frac{\lfloor (n/2) \rfloor}{2} \right\rfloor)$$

$$= 1 + 1 + 1 + C^+(\left\lfloor \frac{\lfloor (\lfloor [n/2] \rfloor /2) \rfloor}{2} \right\rfloor) = \dots$$

Teorema

$$\left\lfloor \frac{\left\lfloor \frac{n}{2^k} \right\rfloor}{2} \right\rfloor = \left\lfloor \frac{n}{2^{k+1}} \right\rfloor$$

$$, \forall n, k \in \mathbb{N}$$

$$C^+(n) = 1 + C^+(\left\lfloor \frac{n}{2} \right\rfloor) = 2 + C^+(\left\lfloor \frac{n}{2^2} \right\rfloor)$$

$$= 3 + C^+(\left\lfloor \frac{n}{2^3} \right\rfloor) = \cdots = \mu + C^+(\left\lfloor \frac{n}{2^\mu} \right\rfloor)$$

Recorrência

$$= 3 + C^{+}(\left\lfloor \frac{n}{2^3} \right\rfloor) = \cdots = \mu + C^{+}(\left\lfloor \frac{n}{2^{\mu}} \right\rfloor)$$

μ é o **menor inteiro** que deve satisfazer ... ?

Recorrência

$$= 3 + C^{+}(\left\lfloor \frac{n}{2^3} \right\rfloor) = \cdots = \mu + C^{+}(\left\lfloor \frac{n}{2^\mu} \right\rfloor)$$

μ é o **menor inteiro** que deve satisfazer

$$\left\lfloor \frac{n}{2^\mu} \right\rfloor < 1$$

$$\frac{n}{2^\mu} < 1$$

Recorrência

$$= 3 + C^+(\left\lfloor \frac{n}{2^3} \right\rfloor) = \cdots = \mu + C^+(\left\lfloor \frac{n}{2^\mu} \right\rfloor)$$

μ é o **menor inteiro** que deve satisfazer

$$\left\lfloor \frac{n}{2^\mu} \right\rfloor < 1$$

$$\frac{n}{2^\mu} < 1$$

$$n < 2^\mu$$

$$\mu > \log_2 n$$

Recorrência

μ é o **menor inteiro** que deve satisfazer

$$\mu > \log_2 n$$

Logo

$$\mu = 1 + \lfloor \log_2 n \rfloor$$

Teorema

O menor inteiro maior que x é

$$\lfloor x \rfloor + 1, \forall x \in \mathbb{R}$$

Pior Caso

$$C^+(n) = \mu + C^+(\left\lfloor \frac{n}{2^\mu} \right\rfloor) = 1 + \lfloor \log_2 n \rfloor + C^+(0) = 1 + \lfloor \log_2 n \rfloor$$

Melhor Caso

De maneira análoga ao pior caso

$$C^{-}(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + min\{C^{-}(\left\lfloor \frac{n-1}{2} \right\rfloor), C^{-}(\left\lfloor \frac{n}{2} \right\rfloor)\}, & \text{se } n > 0 \end{cases}$$

$$C^{-}(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + C^{-}(\left\lfloor \frac{n-1}{2} \right\rfloor), & \text{se } n > 0 \end{cases}$$

$$C^{-}(n) = \lfloor log_2 n \rfloor$$

Melhor Caso

De maneira análoga ao pior caso

$$C^{-}(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + min\{C^{-}(\left\lfloor \frac{n-1}{2} \right\rfloor), C^{-}(\left\lfloor \frac{n}{2} \right\rfloor)\}, & \text{se } n > 0 \end{cases}$$

$$C^{-}(n) = \begin{cases} 0, & \text{se } n = 0, \\ 1 + C^{-}(\left\lfloor \frac{n-1}{2} \right\rfloor), & \text{se } n > 0 \end{cases}$$

$$C^{-}(n) = \lfloor log_2 n \rfloor$$

Obs.: para chegar nesse resultado, uma forma é considerar os casos onde n é par, e onde n é ímpar.

Então

Para uma entrada *C(x,v,a,b)*, ou seja, um vetor de tamanho *n = b - a + 1*, o número de comparações necessárias é

$$\lfloor log_2 n \rfloor \leq C(x,v,a,b) \leq 1 + \lfloor log_2 n \rfloor$$

Pense nisso...

Você criou sua própria rede social, chamada *linkedout*. A rede ficou tão famosa que todos os $7x10^9$ habitantes da terra participam dela.

Considerando que você armazena as pessoas em um vetor ordenado. Quantas comparações são necessárias para se encontrar uma pessoa (digitando seu nome na caixa de buscas) se:

- Você implementar a busca via o algoritmo ingênuo?
- Você implementar a busca via uma busca binária?

Pense nisso...

Você criou sua própria rede social, chamada *linkedout*. A rede ficou tão famosa que todos os $7x10^9$ habitantes da terra participam dela.

Considerando que você armazena as pessoas em um vetor ordenado. Quantas comparações são necessárias para se encontrar uma pessoa (digitando seu nome na caixa de buscas) se:

- Você implementar a busca via o algoritmo ingênuo?
- Você implementar a busca via uma busca binária?
- Como isso reflete nos custos? Na quantidade de hardware necessária? No tempo de resposta da aplicação?

Exercícios

1. Implemente o algoritmo de busca binária recursivo em C
2. Implemente uma versão iterativa do algoritmo de busca binária em C
3. Crie uma função que usa a resposta da busca binária para dizer se x está ou não no vetor, e em qual posição ele está
4. Considere o algoritmo a seguir

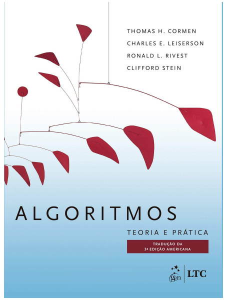
```
função misteriosa (a,b)
entrada: dois números naturais, incluindo o zero, a e b
saída: ???
se b = 0
      retorne 1
se b mod 2 = 0
      retorne misteriosa (a + a, ⌊b/2⌋)
retorne misteriosa (a+a, ⌊b/2⌋) + a
```

   a. O que a função calcula?
   b. Expresse a função como uma recorrência
   c. Qual o número de divisões realizado para uma entrada (a,b) qualquer (Dica: use b como “tamanho” da entrada)?
   d. Qual o número de somas realizado para uma entrada (a,b) qualquer?

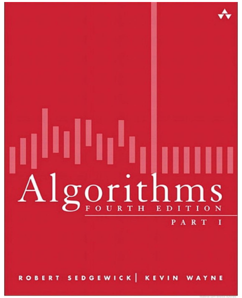
Obs.: no exercício 4 é um tanto estranho usar um número natural como se ele fosse o seu próprio “tamanho”, mas serve como um bom exercício. Uma abordagem melhor, e mais complicada, seria considerar quantos dígitos b possui.

Referências

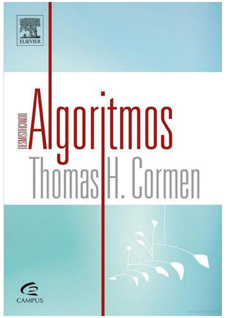
T. Cormen, C. Leiserson, R. Rivest, C. Stein. Algoritmos: Teoria e Prática. 3a ed. 2012



R. Sedgewick, K. Wayne. Algorithms Part I. 4a ed. 2014



T. Cormen. Desmistificando algoritmos. 2017.



H. Schildt. C completo e total. 1996

Licença