Busca Binária

Busca em Vetor Ordenado

Instância: (x, v, a, b), onde x é um valor e $v[a..b]$ é um vetor ordenado.
Resposta: $m \in [a..b]$ tal que $v[m] = x$ ou **não** se tal m não existir.

BuscaBinaria(x,v,a,b)

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

Exemplo. *Executar BuscaBinária*$(8, v, 1, 8)$ *e BuscaBinária*$(20, v, 1, 8)$ *para*

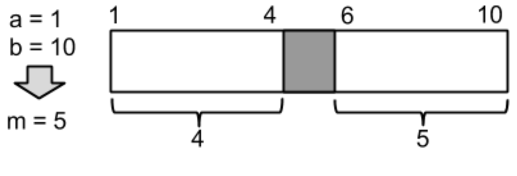
i	1	2	3	4	5	6	7	8
$v[i]$	4	7	8	15	16	17	23	42

Análise do Pior Caso

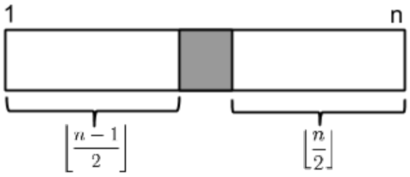
$C^+(n)$: núm. de comparações com elementos do vetor no **pior caso** para entradas de tamanho n.

- OBS: O tamanho das chamadas recursivas dependem se n é par ou ímpar.

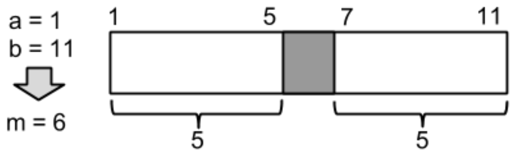
Exemplo para $n = 10$ (par):



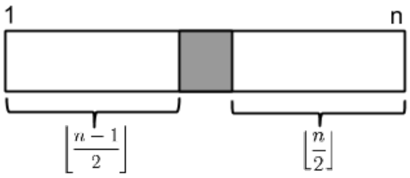
Generalizando, para todo n **par**, o tamanho das chamadas recursivas ficam:



Exemplo para $n = 11$ (ímpar):



Generalizando, para todo n **ímpar**, o tamanho das chamadas recursivas ficam:



Portanto, n sendo par ou ímpar, os tamanhos das chamadas recursivas são $\left\lfloor \frac{n-1}{2} \right\rfloor$ e $\left\lfloor \frac{n}{2} \right\rfloor$.

Assim, podemos escrever:

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

Mas, $\max\left\{C^+\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right), C^+\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right\} = C^+\left(\left\lfloor \frac{n}{2} \right\rfloor\right)$, de forma que

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

Agora podemos resolver a recorrência. Um fato que usaremos como verdade (só será provado em Matemática Discreta):

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

Abrindo a recorrência,

$$\begin{aligned}
C^+(n) =& 1 + C^+\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \\
=& 1 + 1 + C^+\left(\left\lfloor \frac{n}{2^2} \right\rfloor\right) \\
=& 1 + 1 + 1 + C^+\left(\left\lfloor \frac{n}{2^3} \right\rfloor\right) \\
\ldots & \\
=& u + C^+\left(\left\lfloor \frac{n}{2^u} \right\rfloor\right)
\end{aligned}$$

Qual valor de u para chegar em $C^+(0)$? Ou seja, quando $\left\lfloor \frac{n}{2^u} \right\rfloor = 0$?

Isso ocorre quando o denominador é maior que o numerador, ou seja, ocorre para o **menor** u inteiro tal que $2^u > n$.

Isolando u, temos que $u > \log_2 n$. Mas $\log_2 n$ pode ser inteiro ou fracionário:

- Se $\log_2 n$ é inteiro, então $u > \log_2 n$ implica que $u = \log_2 n + 1 = \lfloor \log_2 n \rfloor + 1$.
- Se $\log_2 n$ é fracionário, então $u > \log_2 n$ implica que $u = \lceil \log_2 n \rceil = \lfloor \log_2 n \rfloor + 1$.

Em ambos os casos, o menor u inteiro para chegar na base é $u = \lfloor \log_2 n \rfloor + 1$. Substituindo o valor de u na recorrência:

$$C^+(n) = \lfloor \log_2 n \rfloor + 1 + C^+(0) = \lfloor \log_2 n \rfloor + 1.$$

Assim, podemos enunciar o seguinte Teorema:

Teorema. *Para um vetor de tamanho n, o número de comparações com elementos do vetor do Algoritmo* BuscaBinária *é, no pior caso, $\lfloor \log_2 n \rfloor + 1$.*