Ordenação por inserção binária

Fonte: http://www.php5dp.com/

PF 7.3, 8.1 e 8.2

http://www.ime.usp.br/~pf/algoritmos/aulas/ordena.html

Busca binária

Esta função recebe uma lista crescente $v[0:n]$ com $n \geq 1$ e um inteiro x e retorna um índice j em $\texttt{range}(0, n)$ tal que $v[j] \leq x < v[j{+}1]$

```
def busca_binaria (x, n, v):
    e, d = -1, n
    while e < d-1:   # /*A*/
        m = (e + d) // 2
        if v[m] <= x:   e = m
        else:   d = m
    return e
```

Relações invariantes

A relação invariante chave da função `busca_binaria` é

(i0) Em /*A*/ vale que $v[e] \leq x < v[d]$

A correção da função segue facilmente dessa relação e da condição de parada do `while`.

Busca binária: recordação

O consumo de tempo da função `busca_binaria` é proporcional a $\lg \mathtt{n}$.

O consumo de tempo da função `busca_binaria` é $\mathrm{O}(\lg \mathtt{n})$.

insercao

Função rearranja $v[0:n]$ em ordem crescente.

```
def insercao(v):
    n = len(v)
    for i in range(1,n):   # /*A*/
        x = v[i]
        j = j - 1
        while j >= 0 and v[j] > x:
            v[j+1] = v[j]
            j -= 1
        v[j+1] = x
```

inserca_binaria

Função rearranja $v[0:n]$ em ordem crescente.

```
def insercao_binaria(v):
    n = len(v)
    for i in range(1,n):   # /*A*/
        x = v[i]
        j = busca_binaria(x,i,v)
        for k in range(i,j+1,-1):
            v[k] = v[k-1]
        v[j+1] = x
```

Pior e melhor casos

O maior consumo de tempo da função `insercao_binaria` ocorre quando a lista `v[0:n]` dada é decrescente. Este é o pior caso para a função `insercao_binaria`.

O menor consumo de tempo da função `insercao_binaria` ocorre quando a lista `v[0:n]` dada já é crescente. Este é o melhor caso para a função `insercao_binaria`.

Consumo de tempo no pior caso

linha	consumo de tempo (proporcional a)
0	$= 1$
1	$= n$
2	$= n - 1$
3	$\approx \lg 1 + \lg 2 + \cdots + \lg n \leq n \lg n$
4	$\leq 2 + 3 + \cdots + n = (n-1)(n+2)/2$
5	$\leq 1 + 2 + \cdots + (n-1) = n(n-1)/2$
6	$= n$

total $\leq n^2 + n \lg n + 3n = \mathrm{O}(n^2)$

Consumo de tempo no melhor caso

linha	consumo de tempo (proporcional a)
0	$= 1$
1	$= n$
2	$= n - 1$
3	$\approx \lg 1 + \lg 2 + \cdots + \lg n \leq n \lg n$
4	$= 1 + 1 + \cdots + 1 = n$
5	$= 0$
6	$= n$

total $= n \lg n + 4n = O(n \lg n)$

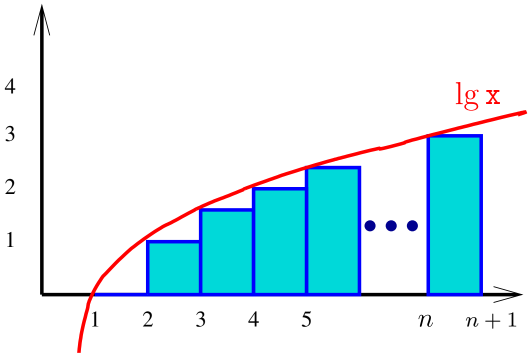
Conclusões

O consumo de tempo da função `insercao_binaria` no pior caso é proporcional a n^2.

O consumo de tempo da função `insercao_binaria` no melhor caso é proporcional a $n \lg n$.

O consumo de tempo da função `insercao_binaria` é $O(n^2)$.

$$\lg 1 + \lg 2 + \cdots + \lg n = O(n \lg n)$$



$$\lg 1 + \lg 2 + \cdots + \lg n \leq \int_1^{n+1} \lg x \, dx$$

$$\lg 1 + \lg 2 + \cdots + \lg n = O(n \lg n)$$

$$\begin{aligned}
\int_1^{n+1} \lg x \, dx &= \Big(\int_1^{n+1} \ln x \, dx \Big) / \ln 2 \\
&= x \ln x - x]_1^{n+1}) / \ln 2 \\
&= ((n+1)\ln(n+1) - (n+1) + 1) / \ln 2 \\
&= ((n+1)\ln(n+1) - n) / \ln 2 \\
&< (n+1)\lg(n+1) \\
&= O(n \lg n)
\end{aligned}$$