Teste 2015/2016 → Parte A ⟵

① pré-condição: $m > 0$

pós-condição: $-1 \leq l < m$

Invariante: $-1 \leq l < m \wedge -1 \leq r \leq m$

Condições de verificação:

// m>0
// I[m/r][-1/l]
l = -1;
// I[m/r]
r = m;
// I
while(...)
	while (l+1 != r) {
	// I ∧ (l+1 != r)
	// I[(l+r)/2/m]
	m = (l + r)/2;
	// I && m == (l+r)/2
	if (a[m] <= x) l = m;
	else r = m;
	}
		while (...) { }
		// I ∧ ¬ (l+1 != r)
		// -1 ≤ l < m

↓ CV1: $m>0 \rightarrow -1 \leq -1 < m \wedge -1 \leq m \leq m$ (inicialização do I)	CV2: $-1 \leq l < m \wedge -1 \leq r \leq m \wedge (l+1 != r) \longrightarrow -1 \leq l < m \wedge -1 \leq r \leq m$ (preservação do I)	CV3: $-1 \leq l < m \wedge -1 \leq r \leq m \wedge \neg(l+1 != r) \longrightarrow -1 \leq l < m$ (utilidade do I)

CV1: $m>0 \rightarrow -1 \leq -1 < m \wedge$ $\wedge -1 \leq m \leq m$ (inicialização de I)	CV2: [illegible] $\rightarrow -1 \leq l < m \wedge -1 \leq r \leq m$ (preservação de I)	[illegible] $\rightarrow -1 \leq l < m$ (utilidade de I)

// I && $m==(l+r)/2$

if (a[m] <= x)

$\quad\quad l = m;$

else

$\quad\quad r = m;$

// $-1 \leq l < m$

CV4: $\models I$ && $m==(l+r)/2 \wedge a[m]<=x \longrightarrow -1 \leq m \leq m$

CV5: $\models I$ && $m==(l+r)/2 \wedge \neg(a[m]<=x) \longrightarrow -1 \leq l < m$

(2)

```
int minInd (char *nomes[], int N){
    int i, r = 0;

    for (i = 1; i < N; i++)
        if (strcmp(nomes[i], nomes[r]) < 0) r = i;

    return r;
}
```

MC da strcmp = strings diferentes na 1ª pos.
PC da strcmp = strings iguais

$\left.\begin{array}{l}\text{MC da minInd}\\ \text{PC da minInd}\end{array}\right\}$ MC = PC, é sempre preciso percorrer todo o array, é necessário ter em atenção o MC e PC da strcmp.

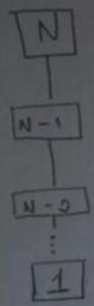
$\boxed{MC}$ $\sum_{i=1}^{N} 1 = n \qquad T(n) = \Theta(n)$

$\boxed{PC}$ $\sum_{i=1}^{N} M = M \times \sum_{i=1}^{N} 1 = M \times (N - 1 + 1)$

$T(n) = \Theta(n^2)$

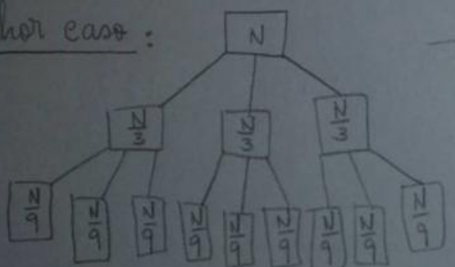
$T(n) = \theta(n$

③ Pior caso:



$$\sum_{i=0}^{N} i = \frac{n(n-1)}{2} \simeq \theta(n^2) = T(n)$$

Melhor caso:



$$T(n) = \theta(n \log_3 n)$$

$$c \times n + 3 \times c \times \frac{n}{3} + 9 \times c \times \frac{n}{9} + 27 \times c \times \frac{n}{27} \ldots$$

→ Parte B ←

```
int intcmp( int n1[], int n2[], int N) {
    int i;
    for( i=N-1 ; (i>=0) && (n1[i]==n2[i]) ; i--);
    if (i<0) return 0;
    else return (n1[i] - n2[i]);
}
```

nº de iterações	quando	probabilidade
1	n1[N-1] ≠ n2[N-1]	$\frac{1}{2}$
2	n1[N-1] == n2[N-1] ∧ n1[N-2] ≠ n2[N-2]	$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$
3	= ∧ = ∧ ≠	$\frac{1}{8}$
x	= ∧ = ∧ =... ∧ ≠	$\underbrace{\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \cdots}_{x-1} \times \frac{1}{2}$ → é diferente

<u>Caso Médio</u>: $\overline{T}(N) = \sum_{x=1}^{N} x \times \left(\frac{1}{2}\right)^{x} \simeq 2 \longrightarrow \Theta(1)$

A complexidade da função minInt é igual ao problema acima, basta mudar para $\sum_{x=1}^{N} x \times \left(\frac{255}{256}\right)^{x} \longrightarrow \Theta(1)$