

Aula 10 15-10-2012

Análise de complexidade

análise
temporalanálise
espacialmemória
acessada

Ficha 2

1) melhor e pior caso

```

int x1search(int a[], int N, int x)
{
    int i;
    i = 0;
    while (i < N && a[i] != x) {
        i++;
    }
    if (i < N) return i;
    else return -1;
}

```

C1

1

C2

m+1

C3

m

C4

1

C5

1

C5

a) / b)

$$T(N) = C_1 + C_2 + C_3 + C_4 + m(C_2 + C_3)$$

Melhor caso $m=0 \Rightarrow$ está na 1ª posição

$$T(N) = C_1 + C_2 + C_3 + C_4 + C_5$$

$$T_N = 4$$

Pior caso $\Rightarrow m = N$ (não encontra)

$$T(N) = C_1 + C_2 + C_3 + C_4 + C_5 + N(C_2 + C_3)$$

simplificando

$$T_N = 4 + 2N$$

$$T(N) = 1 + \sum_{i=0}^{N-1} (1+1) + 1 + 1 + 1 = 4 + 2N$$

```

int lsearch(int a[], int N, int x) //array ordenado
{

```

```

    int i;
    i = 0;

```

```

    while (i < N && a[i] != x)
        i++;

```

```

    if (i < N && a[i] == x) return i;

```

```

    else return -1;
}

```

→ 25 a 203

Melhor caso: encontrar o maior

o pior caso não é o estar (2)
mas sim ser maior do
que todos os elementos