

TP 02

Ana Neri

20 Fevereiro 2023

Exercícios práticos sobre complexidade.

Exercício 1 Considere a função que verifica se um vetor de inteiros contém elementos repetidos.

```
def repetidos(v, N):
    rep = 0
    i = 0
    while i < N-1 and not rep:
        j = i+1
        while j < N and v[i] == v[j]:
            rep = 1
            j += 1
        i += 1
    return rep
```

- Calcule o melhor caso;
- Calcule o pior caso;
- Para o pior caso definido acima, calcule o número de comparações (entre elementos do vetor) que são efetuadas (em função do tamanho do array argumento).

Exercício 2 Considere a função que calcula o número de elementos diferentes num array de inteiros.

```
def diferentes(v, N):
    dif = 0
    for i in range(N):
        j=i+1
        while j<N and v[i]!= v[j]:
            if j == N:
                dif += 1
    return dif
```

- Calcule o melhor caso;
- Calcule o pior caso;
- Para o pior caso definido acima, calcule o número de comparações (entre elementos do vetor) que são efetuadas (em função do tamanho do array argumento).

Exercício 3 Na aula teórica vimos uma função que calculava o produto de dois números.

Considere agora um algoritmo alternativo para este cálculo.

```
def bprod(x, y):
    r = 0
    while x > 0:
        if x and 1:
            r = r + y
        x = x >> 1
        y = y << 1
    return r
```

Qual é o custo da função no pior caso?

Exercício 4 Considere a seguinte definição de uma função que calcula a potência inteira de um número.

```
def pot(base, exp):
    r = 1.0
    while exp > 0:
        r *= base
        exp -= 1
    return r
```

Apresente uma versão alternativa desta função cujo número de multiplicações, no pior caso, seja proporcional ao número de bits usados para representar o expoente (Sugestão: use como inspiração as funções apresentadas na aula teórica e no exemplo anterior para calcular o produto de dois números).

Exercício 5 Considere a seguinte definição da função `inc` que recebe um array de `N` bits (representando um inteiro `x`) e que modifica o array de forma a representar `x+1`.

```
def inc(x, N):
    i = N - 1
    while i >= 0 and x[i] == 1:
        i -= 1
    x[i] = 0
```

```

if i < 0:
    return 1
x[i] = 1
return 0

```

Calcule o custo médio desta função (em termos do número de bits alterados).

Exercício 6 Dado um vetor v de N números inteiros, a mediana do vetor define-se como o elemento do vetor em que

- existem no máximo $N/2$ elementos (estritamente) menores do que ele, e
- existem no máximo $N/2$ elementos (estritamente) maiores do que ele.

Se o vetor estiver ordenado, a mediana corresponde ao valor que está na posição $N/2^1$.

1. Considere a seguinte definição de uma função que calcula a mediana de um vetor com comprimento N ímpar.

```

def mediana(v, N):
    i = -1
    m = N
    M = N
    while i < N and (m > N/2 or M > N/2):
        i = i + 1
        m, M = quantos(v, N, v[i])
    return v[i]

```

Assumindo que a função `quantos` executa em tempo linear no comprimento do vetor de input, identifique o melhor e pior caso de execução da função `mediana`. Para cada um desses casos determine a complexidade assintótica da função `mediana`.

2. Calcule a complexidade média da função apresentada na alínea anterior. Para isso, assumo que os valores do vetor são perfeitamente aleatórios e, por isso, que a probabilidade de o elemento numa qualquer posição do vetor ser a mediana é uniforme ($\frac{1}{N}$).

¹Quando número N é par devemos fazer uma média aritmética entre os 2 números centrais.