



MC438/MC448 - ANÁLISE DE ALGORITMOS
 IC – UNICAMP
 1º Semestre de 2004

1. Dados os algoritmos 1 e 2 para o cálculo do produto de dois números naturais m e n , calcule a complexidade do pior caso. Use para o cálculo da complexidade o número de repetições do loop.

Algoritmo 1.

```
begin
  Z:= 0; u:= n; v:= m;
  while u <> 0 do
    z:= z + v;
    u:= u - 1;
  end;
  write(z);
end;
```

Algoritmo 2.

```
begin
  z:= 0; u:= n; v:= m;
  while u <> 0 do
    begin
      if odd(u) then z:= z + v;
      u:= u div 2;
      v:= 2*v;
    end;
  write(z);
end;
```

2. Considere o algoritmo abaixo que verifica se uma dada sequência C de m caracteres está contida numa sequência T de n caracteres ($m < n$)

```
inicio
  para i:= 1 ate n-m+1 faca
    j:= i; k:= 1;
    enquanto T(j) = C(j) faca
      se k=m entao retorne 'sucesso'
    senao
      inicio
        j:= j + 1
        k:= k + 1
      fim
    retorne 'falha'
  fim
```

Considere como operação básica a comparação entre caracteres.

- Qual é a complexidade no pior caso?
- Supondo que os caracteres possíveis de compor as sequências são apenas x e y , exemplifique uma entrada do tipo pior caso.

3. Calcule o número de somas que o algoritmo abaixo efetua:

```

for i:= 1 to n do
  if odd(n) then
    begin
for j:= i to n do
  x := x + 1;
for j:= 1 to i do
  y:= y + 1;
  end;

```

4. Calcule em função de n , o número de somas e de multiplicações que o algoritmo abaixo efetua

```

for i:= 1 to n/2 do
  begin
    a:= b[i] + c[i];
    for j:= 2*i to n do
      d[j] := a*d[j];
    end;

```

5. Calcule, em função de n o número de multiplicações que o algoritmo abaixo executa:

```

for i:= 1 to n-1 do
  for j:= i+1 to n do
    for k:= 1 to j do
      a[i,k] := j*b[j,k]

```

6. Calcule, em função de n , a complexidade de pior-caso, melhor-caso e caso-médio do procedimento MaxMin abaixo, considerando como operação básica a comparação entre os elementos de um vetor A com n elementos.

```

procedure MaxMin(var A: vetor; var Max, Min: integer);
var i: integer;
begin
  Max := A[1]; Min := A[1];
  for i:= 2 to n do
    if A[i] > Max then Max := A[i]
    else if A[i] < Min then Min := A[i];
  end;

```