Ciências ULisboa | Informática

AED (2017/2018)
Caderno de Exercícios
Folha 1

1 Análise de algoritmos

Cf. Capítulo 4 do livro *Data Structures and Algorithms* do Goodrich&Tamassia.

1. Para cada um dos algoritmos programados abaixo, e assumindo que n é a dimensão do problema, indique o número de operações básicas que são efetuadas no melhor e no pior caso. Analise ainda o caso esperado, considerando que se trata de vectores aleatórios de 0's e 1's.

(a)
```java
boolean exists = false;
int i = 0;
while(!exists && i < n){
    exists = (v[i] == 0);
    i++;
}
```

(b)
```java
boolean exists = false;
int i = 0;
while(!exists && i < n){
    int j = 0;
    while(!exists && j < n){
          exists = (m[i][j] == 0);
          j++;
     }
    i++;
}
```

2. Num dado computador, uma operação demora um milisegundo a ser executada. Considere um programa que realiza um número de operações dado pela função $f(n) = n^5$, para um qualquer parâmetro n. Qual o maior valor de n para o qual o programa é capaz de terminar os seus cálculos se o tempo disponível for (i) um ano, (ii) 10 anos, (iii) 15 mil milhões de anos (a idade do universo)? (um ano tem aproximadamente $3,15 \times 10^7$ segundos).

3. Repita o exercício anterior para as seguintes funções: $f(n) = \log_{10} n$, $f(n) = n/1000$, $f(n) = n/10$, $f(n) = n^{10}$ e $f(n) = 10^n$.

4. Suponha que tem três programas que resolvem o mesmo problema e que fazem, respetivamente, n^2, $\frac{n^2-n}{2}$ e $2n \log n$ operações para resolver uma instância de tamanho n. Supondo que vai correr esses programas num computador em que cada operação demora um milisegundo a ser executada, compare o tempo que os programas gastam para resolver instâncias de tamanho 100, 500, 1000, 5000 e 10000.

5. Suponha que, num determinado computador, o tempo de execução de um determinado programa sobre input de tamanho 1000, 2000, 4000 e 8000 é, respetivamente, 5 segundos, 20 segundos, 80 segundos e 320 segundos.

Estime quanto tempo levará a correr sobre um input de tamanho 16000. Estabeleça uma hipótese relativamente à função que prediz o tempo gasto pelo programa.

6. Mostre que:

   (a) 2004 é $\mathcal{O}(1)$

   (b) $\log(\log(n))$ é $\mathcal{O}(\log n)$ (sugestão: utilize o facto que $\log n$ é $\mathcal{O}(n)$)

   (c) $n^3 \cdot (5n + n^3)$ é $\mathcal{O}(n^6)$

   (d) $(n + 3)^3$ é $\mathcal{O}(n^3)$

   (e) 2^{n+a} é $\mathcal{O}(2^n)$

   (f) $2^{a \cdot n}$ é $\mathcal{O}(2^n)$ se $a < 1$

   (g) 2^n é $\mathcal{O}(n!)$ (sugestão: tabele os primeiros valores de 2^n e $n!$ para descobrir o primeiro n tal que $2^n < n!$; mostre o resultado por indução)

7. Considere $f(n) = 2^n$ e $g(n) = n^n$. Será que f é $\mathcal{O}(g)$ ou g é $\mathcal{O}(f)$? Justifique.

8. Assumindo que T_1 é $\mathcal{O}(f_1)$ e T_2 é $\mathcal{O}(f_2)$, prove as seguintes afirmações.

   (a) $T_1(n) + T_2(n)$ é $\mathcal{O}(\max(f_1(n), f_2(n)))$;

   (b) $T_1(n) \times T_2(n)$ é $\mathcal{O}(f_1(n) \times f_2(n)))$;

   (c) Se T_1 é $\mathcal{O}(f_2)$, então $\mathcal{O}(T_1 + T_2) = \mathcal{O}(f_2)$.

9. Ordene as seguintes funções por taxa de crescimento assintótica: $4n \log n$, 2^{10}, $2^{\log_{10} n}$, $3n + 100 \log n$, 4^n, $n^2 + 10n$.

10. Dê uma caracterização $\mathcal{O}$ do tempo de execução de cada fragmento de código abaixo, assumindo que n é a dimensão de cada problema.

    (a)
    ```
    int m = 0;
    for(int i = 1; i < 10 * n; i++)
      for(int j = 1; j < n; j++)
         m++;
    ```

    (b)
    ```
    int b = n * n;
    while (b > n)
      if (b % 2 == 0)
        b--;
      else b-=2;
    ```

    (c)
    ```
    int soma = 0;
    for(int i = 0; i < n; i++)
      for(int j = 1; j < 4; j++)
        soma += v[i];
    ```

    (d)
    ```
    int x = n * n * n;
    while(x > 1)
      x /= 2;
    ```

(e)
```
int soma = n * n;
while (soma % 2 == 0)
  soma--;
```

(f)
```
for (int i = 0; i < n; i++)
  for (int j = 0; j < n; j++) {
    a[i][j] = 0;
    for (int k = 0; k < n; k++)
      a[i][j] += b[i][k] * c[k][j];
  }
```

(g)
```
int c = 0;
for (int i = 0; i < n; i++)
  for (int j = i; j < n; j++)
    c++;
```

(h)
```
int c = 0;
for (int i = n; i > 0; i/=2)
  for (int j = 0; j < i; j++)
    c++;
```

11. Considere o seguinte algoritmo para o cálculo das médias dos prefixos de um vector. Verifique que o algoritmo dado tem um tempo de execução quadrático e apresente uma solução com tempo linear.

```
Algorithm mediaPrefixos(v,n)
input: um inteiro n e um vector v de números
       de tamanho >= n
output: vector m de tamanho n :
        m[i] é a média dos elementos v[0],...,v[i]

Seja m um vector de números de tamanho n
for i=0 to n-1
  sum:=0
  for j:=0 to i
       sum:=sum+v[j]
  m[i]:=sum/(i+1)
return m
```

12. Recorrendo à análise experimental e ao método científico, estabeleça e valide uma hipótese relativamente à taxa de crescimento assintótica do tempo de execução de cada um dos fragmentos de código abaixo, em função de n.

```
Random rd = new Random();
String s = "";
for (int i = 0; i < n; i++) {
   if (rd.nextInt(2)==0) s += "0";
   else s += "1";
}

Random rd = new Random();
StringBuilder sb = new StringBuilder();
for (int i = 0; i < n; i++) {
   if (rd.nextInt(2)==0) sb.append("0");
   else sb.append("1");
}
String s = sb.toString();
```