Processamento da Informação
– Teoria –

Matrizes

Semana 08

Prof. Jesús P. Mena-Chalco

12/06/2013

Criação de matrizes

```
def criar_matriz_zeros(l,c):
    matriz = [0]*l
    for i in range(0,l):
        matriz[i] = [0]*c
    return matriz
```

Criação de matrizes

```
def criar_matriz_uns(l,c):
    matriz = [1]*l
    for i in range(0,l):
        matriz[i] = [1]*c
    return matriz
```

Menor elemento

Crie uma função que permita determinar o menor elemento de uma matriz dada como parâmetro.

Cabeçalho: def menor_elemento(A):

[[-1,-2,-3,-4,-5,-6], [1,2,3,4,5,6], [7,8,9,10,11,12]] → -6

Menor elemento

```
def menor_elemento(A):
    menor = A[0][0]
    for i in range(0,len(A)):
        for j in range(0,len(A[0])):
            if menor>A[i][j]:
                menor = A[i][j]
    return menor
```

Segundo menor elemento

Crie uma função que permita determinar apenas o segundo menor elemento de uma matriz dada como parâmetro.

Cabeçalho: def segundo_menor_elemento(A)

[[-1,-2,-3,-4,-5,-6], [1,2,3,4,5,6], [7,8,9,10,11,12]] → -5

Segundo menor elemento

```
def segundo_menor_elemento(A):
    menor1 = A[0][0]
    menor2 = A[0][0]
    for i in range(0,len(A)):
        for j in range(0,len(A[0])):
            if menor1>A[i][j]:
                menor2 = menor1
                menor1 = A[i][j]
    return menor2
```

Solução errada!
Teste para

-1	0
1	2

Converter uma matriz em lista

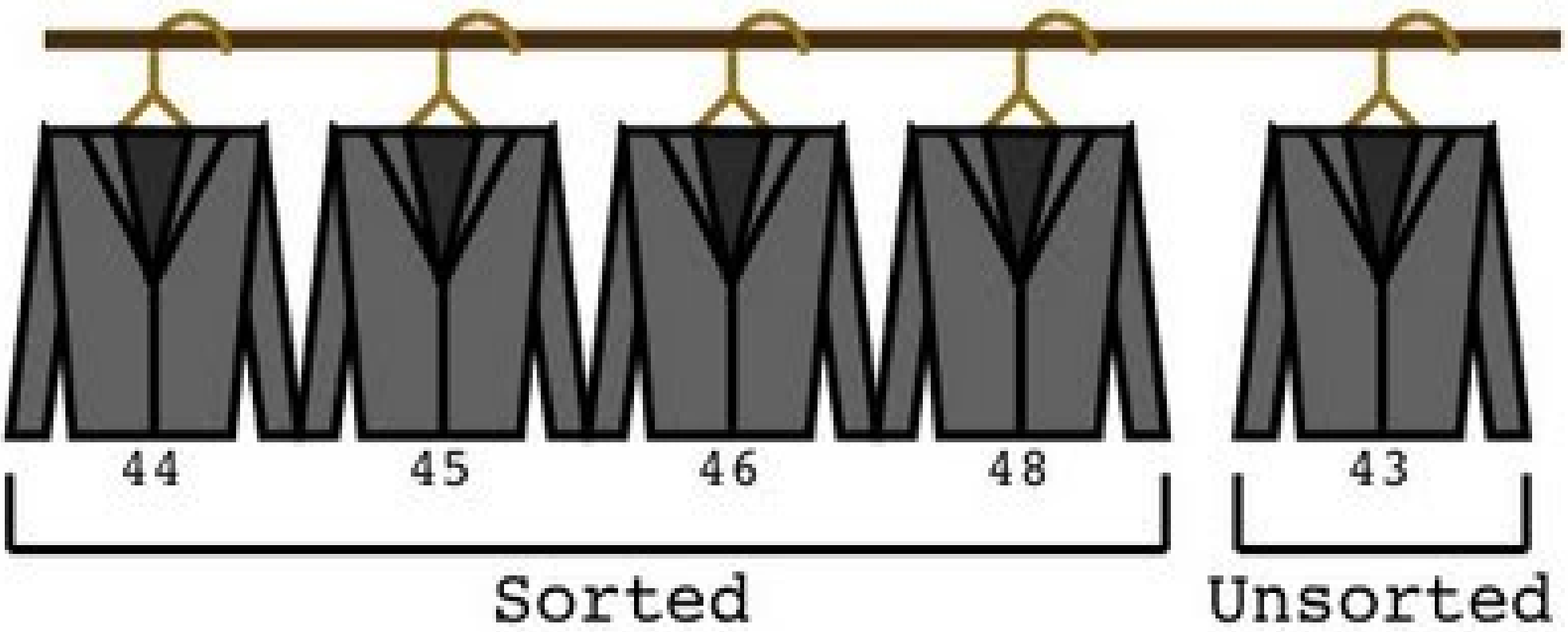
```
def converter_matriz_em_lista(M):
    L = [ ]
    for i in range(0,len(M)):
        for j in range(0,len(M[0])):
            L.append(M[i][j])
    return L
```

Converter uma matriz em lista

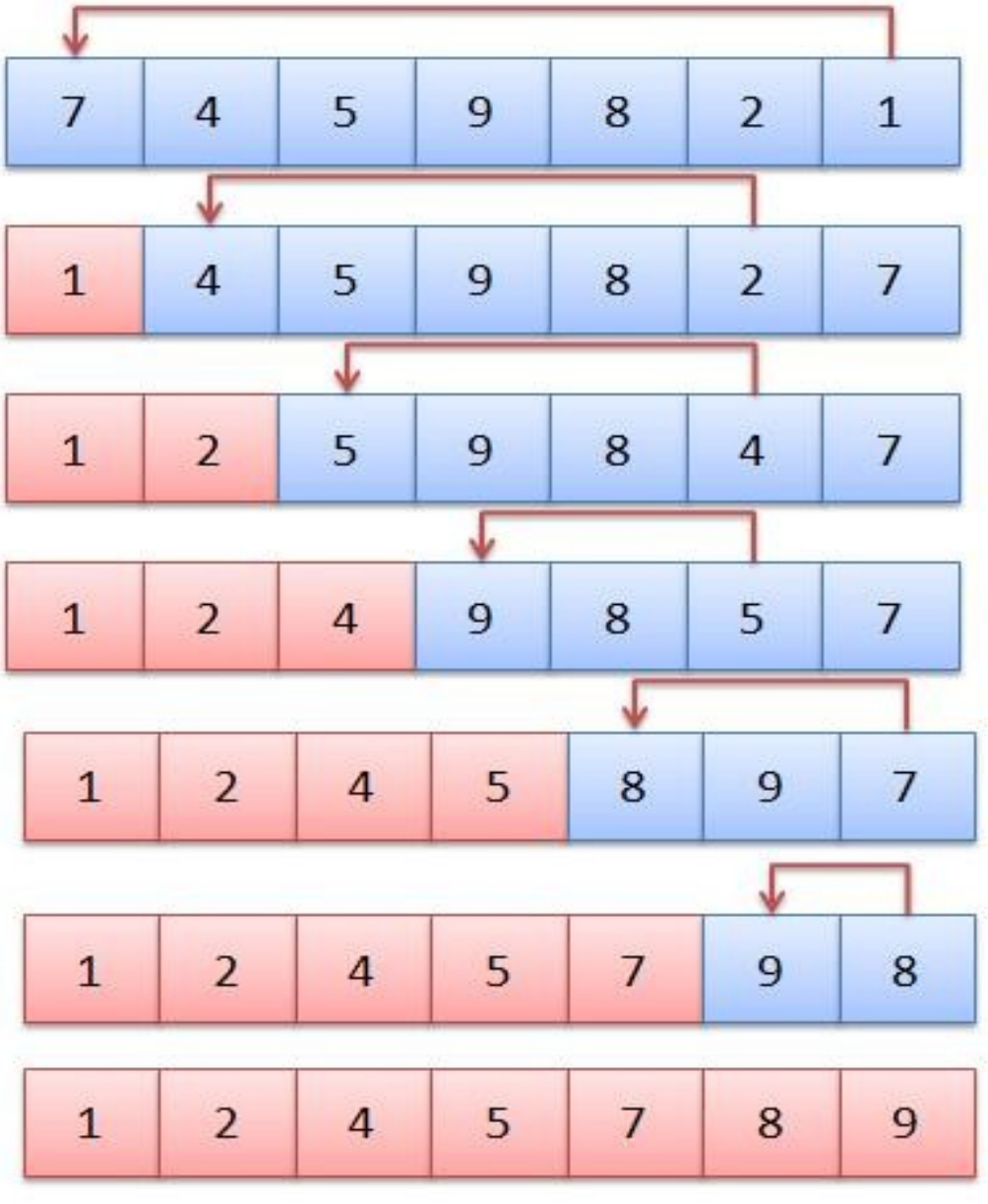
```
def converter_matriz_em_lista(M):
    L = [ ]
    for i in range(0,len(M)):
        for j in range(0,len(M[0])):
            L.append(M[i][j])
    return L
```

```
>>> converter_matriz_em_lista([[1,2],[-1,0],[1,2]])
[1, 2, -1, 0, 1, 2]
```

Ordernar uma lista



Ordernar uma lista



Ordernar uma lista

```
def ordenar_lista(L):
    for i in range(0,len(L)-1):
        menor = i
        for j in range(i+1, len(L)):
            if L[menor]>L[j]:
                menor = j
        if menor!=i:
            temp = L[i]
            L[i] = L[menor]
            L[menor] = temp
    return L
```

Segundo menor elemento

```
def segundo_menor_elemento(A):
    L1 = converter_matriz_em_lista(A)
    L2 = ordenar_lista(L1)
    return L2[1]
```

Segundo menor elemento

```
def segundo_menor_elemento(A):
    L1 = converter_matriz_em_lista(A)
    L2 = ordenar_lista(L1)
    return L2[1]
```

```
>>> segundo_menor_elemento([[1,2],[-1,0],[1,2]])
0
```

ListaOrdenada = [-1, 0, 1, 1, 2, 2]

N-éssimo menor elemento

```
def segundo_menor_elemento(A,n):
    L1 = converter_matriz_em_lista(A)
    L2 = ordenar_lista(L1)
    return L2[n-1]
```

Algoritmos de ordenação?

http://www.sorting-algorithms.com

http://www.youtube.com/watch?v=Ns4TPTC8whw

Matriz triangular superior

Crie uma função que permita verificar se a matriz, dada como parâmetro, é triangular superior.

Cabeçalho: def matriz_triangular_superior(A):

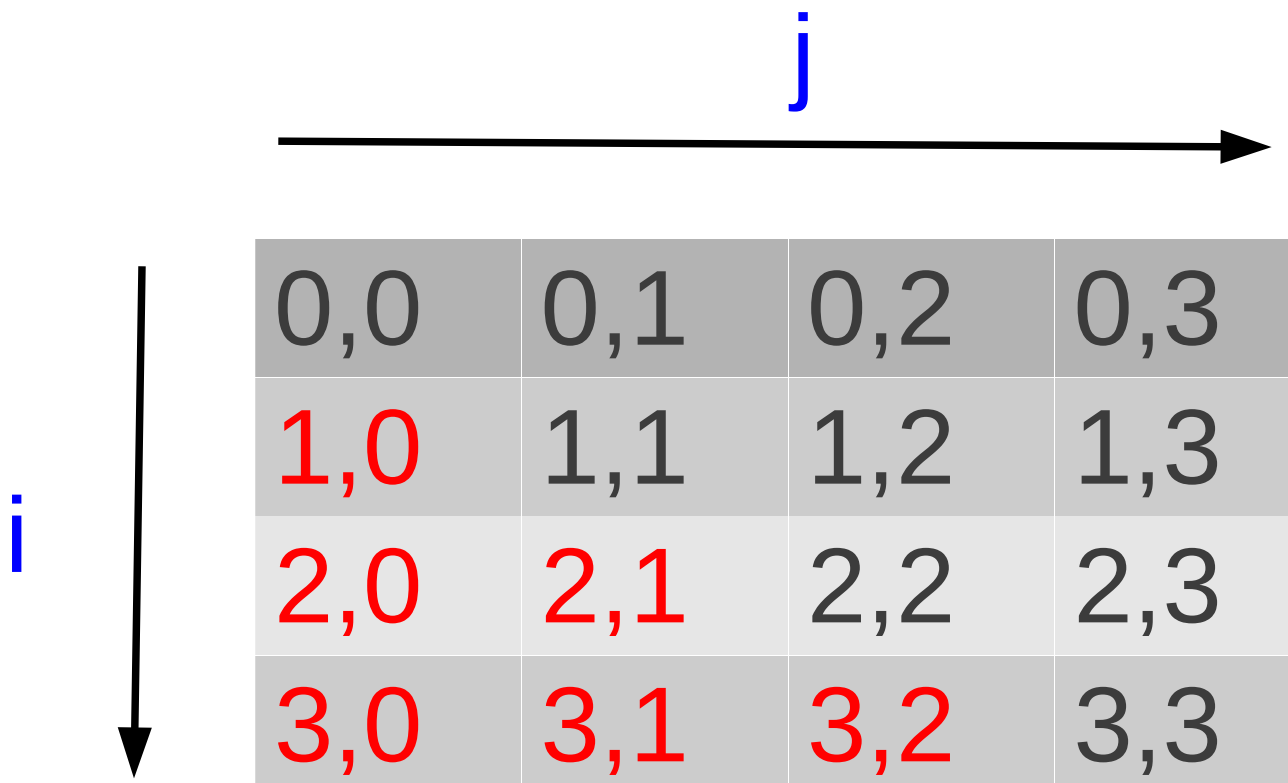
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

Caso particular de matrizes quadradas.

Abaixo da diagonal principal existem apenas elementos nulos.

Os restantes elementos estão posicionados acima dessa mesma diagonal, com a condição de não serem todos nulos.

Matriz triangular superior



0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Índices em uma matriz 4x4

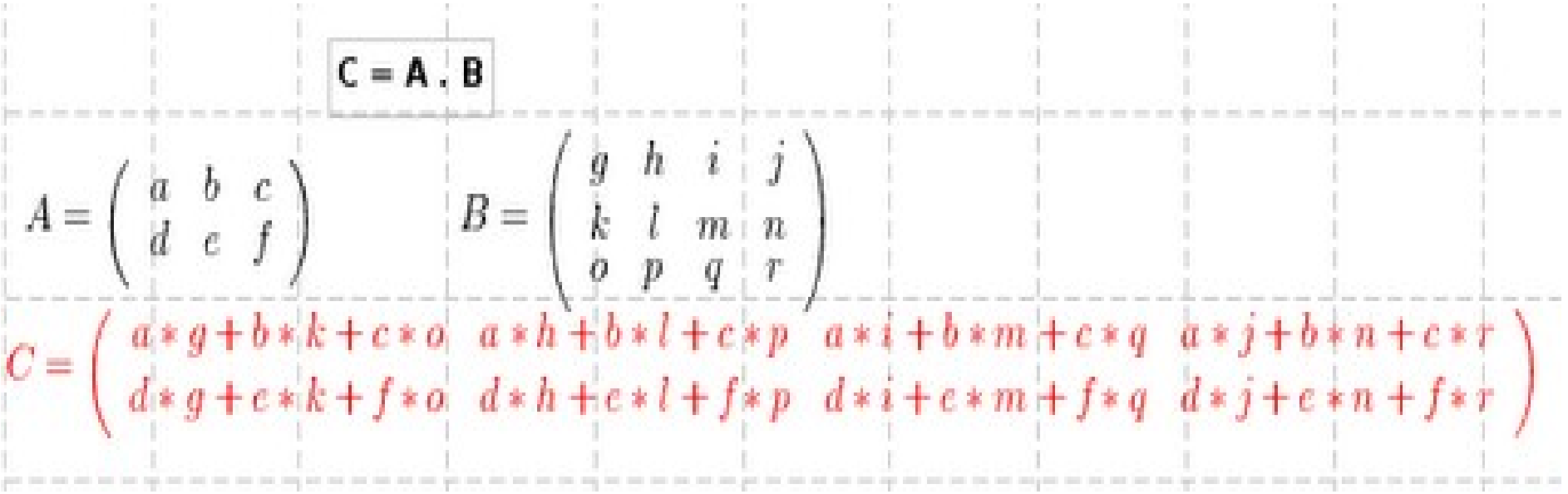
Matriz triangular superior

```
def matriz_triangular_superior(A):
    if len(A[0])!=len(A):
        return False
    contador_zeros_inf = 0
    contador_zeros_sup = 0
    for i in range(0,len(A)):
        for j in range(0,len(A)):
            if i>j and A[i][j]==0:
                contador_zeros_inf +=1
            if i<j and A[i][j]==0:
                contador_zeros_sup +=1
    x = len(A)*(len(A)-1)/2
    if contador_zeros_inf==x and contador_zeros_sup!=x:
        return True
    else:
        return False
```

Multiplicação de matrizes

Crie uma função que permita **multiplicar** duas matrizes dadas como parâmetro.

Cabeçalho: def multiplicar_matrizes(A,B):



Multiplicação de matrizes

```
def multiplicar_matrizes(A,B):
    nlA = len(A)
    ncA = len(A[0])
    nlB = len(B)
    ncB = len(B[0])
    if ncA!=nlB:
        return 'Matrizes com dimensoes incongruentes'
    C = criar_matriz_zeros(nlA,ncB)
    for i in range(0,nlA):
        for j in range(0,ncB):
            val = 0
            for k in range(0,ncA):
                val = val + A[i][k]*B[k][j]
            C[i][j]=val
    return C
```

Multiplicação de matrizes

```
def multiplicar_matrizes(A,B):
    nlA = len(A)
    ncA = len(A[0])
    nlB = len(B)
    ncB = len(B[0])
    if ncA!=nlB:
        return 'Matrizes com dimensoes incongruentes'
    C = criar_matriz_zeros(nlA,ncB)
    for i in range(0,nlA):
        for j in range(0,ncB):
            val = 0
            for k in range(0,ncA):
                val = val + A[i][k]*B[k][j]
            C[i][j]=val
    return C
```

Multiplicação de matrizes

1	1	1
2	2	2

1	2
3	4
5	6

```
>>> multiplicar_matrizes([[1, 1, 1],[2,2,2]],[[1,2],[3,4],[5,6]])
[ [9  , 12],
  [18, 24] ]
```

Lista 05

Questão única.

(a) Dado um inteiro positivo **n**, imprimir as **n** primeiras linhas do triângulo de Pascal(*).

$$
\begin{array}{ccccccc}
1 & & & & & & \\
1 & & 1 & & & & \\
1 & & 2 & & 1 & & \\
1 & & 3 & & 3 & & 1 \\
1 & & 4 & & 6 & & 4 & & 1 \\
1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
\vdots & & & & & & & & & &
\end{array}
$$

(b) Dado um inteiro positivo **n**, imprimir as **n** primeiras linhas do triângulo de Pascal usando apenas uma lista.

(*) Descoberto em 1654 pelo matemático francês Blaise Pascal.

Lista 05

A entrega da Lista 05 deverá ser realizada através do Tidia-ae.

Seção Atividades/lista-05. Até 18/05 (23h50) – Terça-feira.

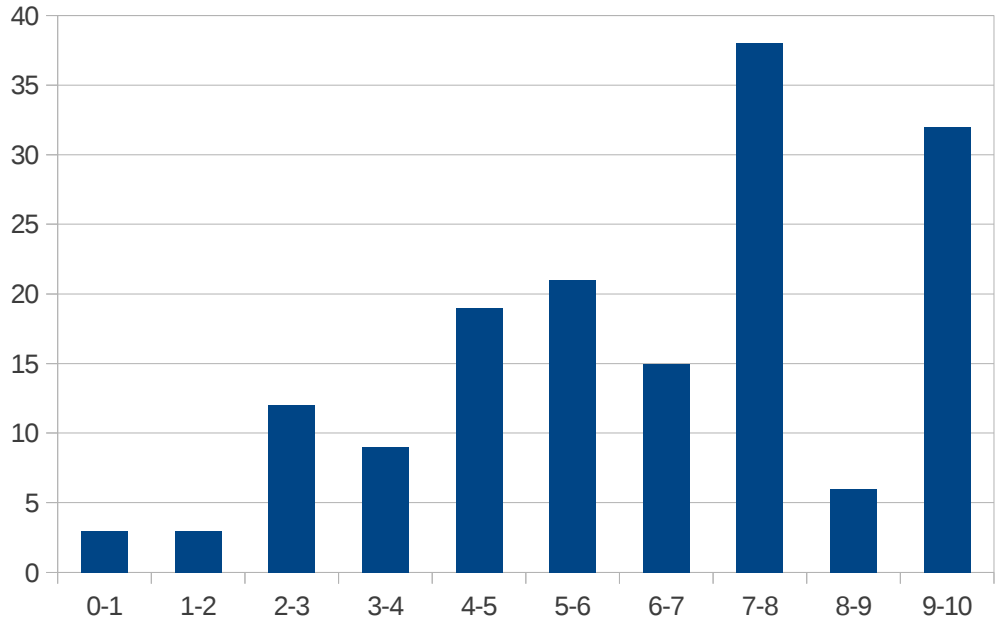
Apenas deve ser enviado um arquivo PDF contendo a solução das questões. O documento deve ter o seguinte nome: RA-SeuNomeCompleto-Lista-05.pdf

Sobre a Prova 01

Alunos: 159

Média: 6,66

Aprovados: 112 (~70%)
Reprovados: 47 (~30%)



O gabarito está no caderno de exercícios:
http://professor.ufabc.edu.br/~jesus.mena/courses/bc0505-1