

THỰC HÀNH MÔN TOÁN RỜI RẠC

| | |
|--------------------|---|
| Thời gian | : 3 buổi |
| Ngôn ngữ lập trình | : Python , Java, C++ |
| Phần mềm | : Visual Studio Code , Eclipse, CodeBlocks |
| Yêu cầu | : SV nắm bài trên lớp, có ý tưởng / sơ đồ khối của các thuật toán |
| Đánh giá | : 10% ĐQT. |

Chương 1. Các phép đếm cơ bản

1. Nhân ma trận. Ví dụ:
$$\begin{bmatrix} 2 & 0 & -1 \\ 1 & 3 & -2 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 1 & 0 \\ 2 & 3 & -1 & 4 \\ -3 & 0 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -2 & 4 & -1 \\ 12 & 8 & 2 & 10 \end{bmatrix}$$

Cách 1:

```
1 import numpy as np
2 A = [[ 2,  0, -1],
3      [ 1,  3, -2]]
4 B = [[ 0, -1,  1,  0],
5      [ 2,  3, -1,  4],
6      [-3,  0, -2,  1]]
7 np.dot(A, B)
```

Cách 2:

```
1 def matrix_mul(A, B):
2     m, n = len(A), len(B)
3     p = len(B[0])
4     C = [[0 for j in range(p)] for i in range(m)]
5     for i in range(m):
6         for j in range(p):
7             for k in range(n):
8                 C[i][j] += A[i][k] * B[k][j]
9     return C
10 A = [[ 2,  0, -1],
11      [ 1,  3, -2]]
12 B = [[ 0, -1,  1,  0],
13      [ 2,  3, -1,  4],
14      [-3,  0, -2,  1]]
15 matrix_mul(A, B)
```

2. Liệt kê các xâu nhị phân độ dài n . Ví dụ: các xâu nhị phân độ dài 3 gồm 000, 001, 010, 011, 100, 101, 110, 111.

Cách 1:

```
1 import itertools
2 A = ['0', '1']
3 list( itertools.product(A, A, A) )
```

Cách 2:

```
1 def binary_strs(n):
2     if n==1:
3         return ['0', '1']
4     S = []
5     for s in binary_strs(n-1):
6         S.append('0' + s)
7     for s in binary_strs(n-1):
8         S.append('1' + s)
9     return S
10 binary_strs(3)
```

3. Thuật toán sắp xếp nổi bọt

```
1 def BubbleSort(a):
2     n = len(a)
3     for i in range(n-1):
4         for j in range(n-1,i,-1):
5             if a[j] < a[j-1]:
6                 a[j-1], a[j] = a[j], a[j-1]
7     return a
8 BubbleSort([7, 9, 2, 5, 8])
```

4. Tính $n! = 1 \cdot 2 \cdots n = n \cdot (n-1)!$. Ví dụ: $3! = 6$

Cách 1:

```
1 from sympy import *
2 factorial(3)
```

Cách 2:

```
1 def factorial(n):
2     if n == 0:
3         return 1
4     return n * factorial(n-1)
5 P(3)
```

Cách 3:

```
1 factorial = lambda n: 1 if n==0 else n * factorial(n-1)
```

Cách 4:

```
1 def factorial(n):
2     p = 1
3     for i in range(1, n+1):
4         p *= i
5     return p
```

5. Liệt kê các hoán vị của n vật. Ví dụ: các hoán vị của ba số 1, 2, 3 là 123, 132; 213, 231; 312, và 321.

Cách 1:

```
1 import itertools
2 list( itertools.permutations([1, 2, 3]) )
```

Cách 2:

```
1 def permutations(a):
2     n = len(a)
3     if n==1:
4         return [a]
5     P = []
6     for i in range(n):
7         b = a.copy()
8         b.pop(i)
9         for p in permutations(b):
10             p = [a[i]] + p
11             P.append(p)
12     return P
13 permutations([1, 2, 3])
```

6. Tính $P(n, r) = n(n-1)\cdots(n-r+1) = \frac{n!}{(n-r)!}$. Ví dụ: $P(5, 3) = 60$.

Cách 1:

```
1 from sympy import *
2 n, r = 5, 3
3 factorial(n) / factorial(n-r)
```

Cách 2:

```
1 def P(n, r):
2     prod = 1
3     for i in range(n, n-r, -1):
4         prod *= i
5     return prod
6 P(5, 3)
```

7. Liệt kê các chỉnh hợp chập r của n vật. Ví dụ: các chỉnh hợp chập 3 của các số 1, 2, 3, 4 là 123, 124, 132, 134, 142, 143; 213, 214, 231, 234, 241, 243; 312, 314, 321, 324, 341, 342; 412, 413, 421, 423, 431, 432.

Cách 1:

```
1 import itertools
2 list(itertools.permutations([1, 2, 3, 4], 3))
```

Cách 2:

```
1 def permutations(a, r):
2     if r == 1:
3         return [[i] for i in a]
4     P = []
5     n = len(a)
6     for i in range(n):
7         b = a.copy()
8         b.pop(i)
9         for p in permutations(b, r-1):
10             p = [a[i]] + p
11             P.append(p)
12     return P
13 permutations([1, 2, 3, 4], 3)
```

8. Tính $\binom{n}{r} = \frac{n(n-1)\cdots(n-r+1)}{r!} = \frac{n!}{r!(n-r)!}$. Ví dụ: $\binom{5}{3} = 10$

Cách 1:

```
1 from sympy import *
2 n, r = 5, 3
3 factorial(n) // factorial(r) // factorial(n-r)
```

Cách 2:

```
1 n, r = 5, 3
2 c = 1
3 for i in range(r):
4     c *= (n - i) // (i + 1)
5 c
```

Cách 3: $\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}$ với $n > r \geq 1$, trong đó $\binom{n}{0} = \binom{n}{n} = 1$

```
1 def binomial(n, r):
2     if r==0 or r==n:
3         return 1
4     return binomial(n-1, r) + binomial(n-1, r-1)
```

Cách 4:

```
1 binomial = lambda n, r: 1 if r==0 or r==n else binomial(n-1, r)
   + binomial(n-1, r-1)
```

Cách 5:

```
1 def pascal_row(n):
2     if n==0:
3         return [1]
4     a = pascal_row(n-1)
5     for i in range(n-1, 0, -1):
6         a[i] += a[i-1]
7     a.append(1)
8     return a
9 pascal(5)[3]
```

9. Liệt kê tổ hợp chập r của n vật. Ví dụ: các tổ hợp chập 3 của các số 1, 2, 3, 4, 5 là 123, 124, 125, 134, 135, 145; 234, 235, 245, 345.

Cách 1:

```

1 import itertools
2 list( itertools.combinations([1, 2, 3, 4, 5], 3) )

```

Cách 2:

```

1 def combinations(a, r):
2     if r == 1:
3         return [[i] for i in a]
4     if r == len(a):
5         return [a]
6     C = []
7     for c in combinations(a[1:], r-1):
8         C.append([a[0]] + c)
9     for c in combinations(a[1:], r):
10        C.append(c)
11    return C
12 combinations([1, 2, 3, 4, 5], 3)

```

10. (*) Liệt kê các hoán vị lặp của n_1 vật loại 1, ..., n_r vật loại r . Ví dụ: các hoán vị lặp của chữ COOL là CLOO, COLO, COOL, LCOO, OCLO, OCOL, LOCO, OLCO, OOLC, LOOC, OLOC, OOLC

11. Liệt kê các nghiệm nguyên không âm của phương trình $x_1 + x_2 + \dots + x_n = r$. Ví dụ: các nghiệm nguyên không âm $x_1 x_2 x_3$ của phương trình $x_1 + x_2 + x_3 = 4$ là 004, 013, 022, 031, 040, 103, 112, 121, 130, 202, 211, 220, 301, 310, 400

```

1 def solutions(n, r):
2     if n == 1:
3         return [[r]]
4     S = []
5     for i in range(r+1):
6         for s in solutions(n-1, r-i):
7             s = [i] + s
8             S.append(s)
9     return S
10 solutions(3, 4)

```