目录

Listing 1: python code

```python
import numpy as np

tp = (1,2,3,4,5,6,7,8,9,10)
print(tp[:5])
print(tp[5:])
"string"

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

```

```
34      return M
```

Listing 2: python code

```
1   import numpy as np
2
3   def incmatrix(genl1,genl2):
4       m = len(genl1)
5       n = len(genl2)
6       M = None #to become the incidence matrix
7       VT = np.zeros((n*m,1), int)  #dummy variable
8
9       #compute the bitwise xor matrix
10      M1 = bitxormatrix(genl1)
11      M2 = np.triu(bitxormatrix(genl2),1)
12
13      for i in range(m-1):
14          for j in range(i+1, m):
15              [r,c] = np.where(M2 == M1[i,j])
16              for k in range(len(r)):
17                  VT[(i)*n + r[k]] = 1;
18                  VT[(i)*n + c[k]] = 1;
19                  VT[(j)*n + r[k]] = 1;
20                  VT[(j)*n + c[k]] = 1;
21
22                  if M is None:
23                      M = np.copy(VT)
24                  else:
25                      M = np.concatenate((M, VT), 1)
26
27                  VT = np.zeros((n*m,1), int)
28
29      return M
```