

{% note info %} **摘要** Title: 861. 二分图的最大匹配 Tag: 匈牙利算法、二分图 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

861. 二分图的最大匹配

- 题意

给定一个二分图，其中左半部包含 n_1 个点（编号 $1 \sim n_1$ ），右半部包含 n_2 个点（编号 $1 \sim n_2$ ），二分图共包含 m 条边。数据保证任意一条边的两个端点都不可能在同一部分中。请你求出二分图的最大匹配数。

- 思路

二分图的匹配：给定一个二分图 G ，在 G 的一个子图 M 中， M 的边集 $\{E\}$ 中的任意两条边都不依附于同一个顶点，则称 M 是一个匹配。二分图的最大匹配：所有匹配中包含边数最多的一组匹配被称为二分图的最大匹配，其边数即为最大匹配数。

匈牙利算法，实际运行远小于 $O(n*m)$

- 代码

```
...
Author: NEFU AB-IN
Date: 2022-03-04 22:36:48
FilePath: \ACM\Acwing\861.py
LastEditTime: 2022-03-05 11:27:14
...
N = 510

match, st = [0] * N, [0] * N # match 表示左边选择的右边点
g = [[] for _ in range(N)]

def find(u):
    for v in g[u]:
        if st[v] == 0:
            st[v] = 1
            if match[v] == 0 or find(match[v]): #如果右边未匹配，或者与右相连的
左点有别的匹配
                match[v] = u
            return True
    return False
```

```
n1, n2, m = map(int, input().split())
res = 0

for i in range(m):
    u, v = map(int, input().split())
    g[u].append(v)

for i in range(1, n1 + 1):
    st = [0] * N # 每次清空st数组, 表示右边目前都未考虑过, 相当于回溯
    if find(i):
        res += 1
print(res)
```