

算法设计与分析

李恒武

smxlhw@sina.com

第一章 算法与问题



- 1.1 稳定匹配问题
- 1.2 Hi! 什么是算法?
- 1.3 大学入学申请问题



重点和难点

□ 教学重点

- 问题求解

□ 教学难点

- 问题求解

- 正确性证明

稳定匹配问题

- ❑ 小波的爸爸GS开了一个婚姻介绍所，会员有 n 个单身男孩 n 个单身女孩。爸爸想让小波设计一个程序，快速配对成 n 对新人，并且使男孩女孩都比较满意，比较稳定。
- ❑ 问题分析：输入输出—设计算法
复杂模糊—清晰的描述 透过现象看本质
数学描述—数学模型

稳定匹配问题



□ 输入？ 现实的模拟？

- n 个单身男孩 n 个单身女孩
- 男孩给女孩打分排序， 女孩给男孩打分排序
 - ✓ 男孩的喜欢列表
 - ✓ 女孩的喜欢列表

男孩	1 st	2 nd	3 rd
X	A	B	C
Y	B	A	C
Z	A	B	C

女孩	1 st	2 nd	3 rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z



稳定匹配问题



- 输出?
- n 对新人 $X-A, Y-B, Z-C$
 - 一夫一妻制
 - 每一个男孩正好和一个女孩是一对.
 - 每一个女孩正好和一个男孩是一对.
- 完美匹配: 每一个人有且仅有一个配对.
- 怎么保证比较满意比较稳定呢?



稳定匹配问题



□ Q. X-C, Y-B, Z-A 稳定? X-B?

	1 st	2 nd	3 rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1 st	2 nd	3 rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

稳定匹配问题



- Q. X-C, Y-B, Z-A 稳定?
- A. 不. B-X不稳定——称为不稳定配对.

	1 st	2 nd	3 rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1 st	2 nd	3 rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

稳定匹配问题



□ Q. X-A, Y-B, Z-C 稳定? Z-B? Z-A?

□ A. Yes. 单相思接受.

	1 st	2 nd	3 rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1 st	2 nd	3 rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

问题分析



- 输入：两张喜欢列表
- 输出：一个稳定匹配-没有不稳定配对的完美匹配。
- 稳定匹配问题. 给定 n 男孩 和 n 个女孩的喜欢列表, 是否存在一个稳定匹配, 如果是找出稳定匹配。

新问题?

- 怎样有效的找出这个稳定匹配?





男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

初始化所有人单身

男孩根据喜欢列表从高到低约会



男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

W约会D
- D单身



男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

W约会D

- D单身，所以同意了
- W-D成为一对

男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

X约会B
- B单身



男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

X约会B

- B单身

-X-B成为一对

男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

Y约会A
- A单身

男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

Y约会A

- A单身

-Y-A成为一对



男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

Z约会B

- B更喜欢现在的男朋友X



男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

Z约会B

- B更喜欢现在的男朋友X
- Z被拒绝



男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

Z约会D

- D更喜欢Z，甚于男朋友W



男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

Z约会D

-Z-D成为一对

-W成为单身

男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

W约会B

- B更喜欢男朋友X

男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

W约会B
- 被拒绝



男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

W约会A

- A更喜欢男朋友Y

男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

W约会A

- 又被拒绝

男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

W约会C
- C单身

男	0	1	2	3
W	D	B	A	C
X	B	C	D	A
Y	A	D	C	B
Z	B	D	A	C

女	0	1	2	3
A	Z	X	Y	W
B	X	W	Y	Z
C	W	X	Y	Z
D	Z	Y	X	W

W约会C

-W-C成为一对

结果: AY BX CW DZ



□ [Gale-Shapley 1962] 保证找到一个稳定匹配。 - 算法思想

初始化每个人都单身。

1. 一个单身男孩沿着喜欢列表从高到低选择女孩。

如果 w 是单身，接受

如果 w 已配对，但更喜欢 m 甚于目前的男友，则抛弃男友，与 m 约会

如果 w 已配对，但更喜欢目前的男友甚于 m ，则拒绝 m 。

2. 转至1，直至没有单身男人为止。

正确性证明



□ 正确性证明

- 算法在有限时间终止
- 算法结果是一个稳定匹配 没有不稳定配对的完美匹配



正确性证明: 算法终止



□ **声明: 算法在至多 n^2 次循环后终止。**

■ **证明: 每一次循环中, 都有一个男孩向一个他还没有约会过的女孩发出邀请。 每个男孩至多约会 n 次, 共有 n^2 种可能。 ▀**

■ **最后一个单身女孩接受约会时算法终止**

	1 st	2 nd	3 rd	4 th	5 th
Vi	A	B	C	D	E
W	B	C	D	A	E
X	C	D	A	B	E
Y	D	A	B	C	E
Z	A	B	C	D	E

	1 st	2 nd	3 rd	4 th	5 th
A	W	X	Y	Z	V
B	X	Y	Z	V	W
Ci	Y	Z	V	W	X
Di	Z	V	W	X	Y
E	V	W	X	Y	Z

$n(n-1) + 1$ 次约会



- 观察 1. 男孩沿着喜欢列表从高到低约会女孩。
- 观察 2. 一旦一个女孩有男友，会一直有男友。



正确性证明: 完美匹配



□ 所有的男孩和女孩最终都一一配对.完美匹配

■ Pf. (反证)

- 假定存在男孩 **Z** 在算法终止时单身。
- 那么必存在一个女孩在算法终止时单身, 设为 **A**。
- 因为 **Z** 最后单身, 所以他已经向每一个女孩发出过约会邀请, 否则算法不会终止。
- 约会 **A**, **A** 接受, 最终配对。
 A 拒绝 (当时或以后), **A** 最终配对。



正确性证明: 稳定性



□ Claim. 没有不稳定配对.

■ Pf. (反证)

➤ 假定 A-Z 是不稳定配对。两种情况

↙ 男孩根据喜欢程度从高到低约会

➤ Case 1: Z 从没向 A 发出过约会邀请.

⇒ Z 喜欢现在的女朋友甚于 A.

⇒ A 和 Z 是稳定的.

S^*
A-Y
B-Z
...



正确性证明: 稳定性



□ Claim. 没有不稳定配对

■ Pf. (反证)

- Case 2: Z 向 A 发出过约会邀请.
 - ⇒ A 拒绝 Z (当时或以后)
 - ⇒ A 喜欢现在的男朋友甚于 Z.
 - ⇒ A 和 Z 都是稳定的.

A-Y
B-Z
...

- 在任何情形, A 和 Z 都是稳定的. 与假设矛盾.



小结

- 稳定匹配问题：给定 n 男孩 和 n 个女孩的喜欢列表，是否存在一个稳定匹配，如果是找出稳定匹配。
- Gale-Shapley 算法：对于任意喜欢列表，保证找到一个稳定匹配。

新问题

- Q. 是否有多个稳定匹配？
- Q. 如果存在多个稳定匹配，GS算法发现的是哪一个？



- ❑ 存在两个稳定匹配
- ❑ GS找到的是第一个匹配

男孩	1 st	2 nd
X	A	B
Y	B	A

女孩	1 st	2 nd
A	Y	X
B	X	Y

算法特性



- ❑ GS每次执行都产生对男孩最优的匹配
 - ❑ 完美 稳定.同时对于男孩最好，对于女孩最差
- 新问题？
- ❑ 算法思想如何结合数据结构细化？
 - ❑ 如何实现算法？





□ 表示男孩和女孩.

■ 男孩 $0, \dots, n-1$.

■ 女孩 $0, \dots, n-1$.

□ 两张喜欢列表—二维数组 $\text{boy}_{n \times n}$ $\text{girl}_{n \times n}$





□ [Gale-Shapley 1962] 保证找到一个稳定匹配。▶

1. 初始化每个人都单身.

2. **while** (某个男孩单身 m , 并且他有没约会的女孩) **do** $\text{boy}[m][\text{count}[m]]$

3. 选择这个男孩 m 他没约会过的女孩中最喜欢的 $w = 1^{\text{st}}$

4. **if** (w 单身)

5. **then** m 和 w 成为一对

6. **else if** (w 喜欢 m 甚于现在的男朋友 m')

7. **then** m 和 w 成为一对, 并且 m' 成为单身

8. **else** w 拒绝 m

$\text{wife}[i] = -1, \text{husband}[j] = -1$

单身男孩 栈free 动态维护

$\text{wife}[m] = w, \text{husband}[w] = m$

$\text{invertGirl}_{n \times n}$



□ 约定.

- 一个未婚男孩的列表, 使用队列或堆栈
- 两个数组 $wife[m]$, $husband[w]$.
 - 未婚0
 - 如果 m 与 w 配对, $wife[m]=w$ 且 $husband[w]=m$

□ 男孩约会的女孩.

- 每一个男孩, 有一个对女孩的排序表。
- 数组 $count[m]$ 计数 m 约会数, $boy[m][count[m]]$ 就是未约会过的最喜欢的女孩。





□ [Gale-Shapley 1962] 保证找到一个稳定匹配。▶

1. 初始化每个人都单身.

2. **while** (某个男孩单身 m , 并且他有没求过婚的女孩) $\leftarrow \text{boy}[m][\text{count}[m]]$

3. 选择这个男孩 m 他没求过婚的女孩中最喜欢的 $w = 1^{\text{st}}$

4. **if** (w 单身)

5. **then** m 和 w 成为一对

6. **else if** (w 喜欢 m 甚于现在的男朋友 m')

7. **then** m 和 w 成为一对, 并且 m' 成为单身

8. **else** w 拒绝 m

$\text{wife}[i] = -1, \text{husband}[j] = -1$

栈或队列 free 动态维护

$\text{boy}[m][\text{count}[m]]$

$\text{wife}[m] = w, \text{husband}[w] = m$

$\text{invertGirl}_{n \times n}$

$\text{wife}[m'] = -1, \text{husband}[w] = m$



■ 女孩拒绝/接受.

- 女孩 w 喜欢 m 甚于 m' ?
- 每一个女孩, 根据喜欢列表产生一个喜欢程度列表 `inverse`。
- $O(n)$ 预处理后每次查询仅用常数时间。

Amy	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th
girl	8	3	7	1	4	5	6	2

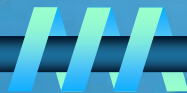
Amy 喜欢男孩3 甚于 6
`inverse[3] < inverse[6]`

Amy	1	2	3	4	5	6	7	8
Inverse	4 th	8 th	2 nd	5 th	6 th	7 th	3 rd	1 st

2

7

```
for i = 1 to n
    inverse[girl[i]] = i
```



```
while(Stack.size()!=0) {  
    int man = Stack.top();    Stack.pop(); //选择单身男孩man  
    dating(Stack, man, boy, girl, prefer, wife, count, husband); //与prefer约会  
}  
  
void dating(stack<int>& Stack, int man, int boy[][NUM], int girl[][NUM], int wife[], int count[],  
            int husband[]) {  
    int prefer= boy[man][count[man]]; //选择未约会排首位的女孩  
    if(husband[prefer] == NIL) { husband [prefer] = man; wife[man] = prefer; //女孩单身, 则接受.  
    else { int old = invertGirl[prefer][husband[prefer]];  
           int new= invertGirl[prefer][man];  
           if(old< new) { count[man]++; Stack.push(man); } //被拒绝, 加入单身栈  
           else {count[husband[perfer]]++; //甩掉男友  
                  Stack.push(husband[perfer]); //被甩的男友, 加入单身栈  
                  husband[perfer] = man; wife[man] = perfer; } } } //换男友
```

稳定匹配总结

- ❑ 稳定匹配问题：给定 n 男孩 和 n 个女孩的喜欢列表，是否存在一个稳定匹配，如果是找出稳定匹配。
- ❑ Gale-Shapley 算法：对于任意实例，保证找到一个稳定匹配。
 - n^2 次数量级计算.
 - 男孩最优.

问题的求解

