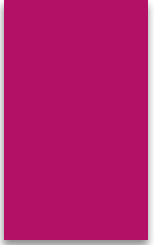




Problem Analysis Of Stable Match

YAO ZHAO



Initially all $m \in M$ and $w \in W$ are free

While there is a man m who is free and hasn't proposed to every woman w for which $(m, w) \notin F$

 Choose such a man m

 Let w be the highest-ranked woman in m 's preference list to which m has not yet proposed

 If w is free then

(m, w) become engaged

 Else w is currently engaged to m'

 If w prefers m' to m then

m remains free

 Else w prefers m to m'

(m, w) become engaged

m' becomes free

 Endif

Endif

Endwhile

Return the set S of engaged pairs

Common Problems

What data structures are used for input/output?

How to find unmatched men efficiently?

How to efficiently query the ranking of a man in a woman's preference list?

How to test the code sufficiently?

What data structures are used for input/output?

Men's Preference Profile

	0 th	1 st	2 nd	3 rd	4 th
Victor	Bertha	Amy	Diane	Erika	Clare
Wyatt	Diane	Bertha	Amy	Clare	Erika
Xavier	Bertha	Erika	Clare	Diane	Amy
Yancey	Amy	Diane	Clare	Bertha	Erika
Zeus	Bertha	Diane	Amy	Erika	Clare

Women's Preference Profile

	0 th	1 st	2 nd	3 rd	4 th
Amy	Zeus	Victor	Wyatt	Yancey	Xavier
Bertha	Xavier	Wyatt	Yancey	Victor	Zeus
Clare	Wyatt	Xavier	Yancey	Zeus	Victor
Diane	Victor	Zeus	Yancey	Xavier	Wyatt
Erika	Yancey	Wyatt	Zeus	Xavier	Victor

How to find unmatched men efficiently?

- ▶ Initial, all men are free and added to a queue
- ▶ In each iteration, get a free man, try to match; if a woman prefers this man over her current provisional partner, the woman will dump her current provisional partner, who becomes free again and is added to the queue.

How to find a man's next preferred woman after being rejected?

In the following case, Victor is dumped by Bertha, go back to queue. We can record the index of Bertha. When he is popped from queue again, he can propose to Amy(the index of Bertha+1).

Men's Preference Profile

	0 th	1 st	2 nd	3 rd	4 th
Victor	Bertha	Amy	Diane	Erika	Clare
Wyatt	Diane	Bertha	Amy	Clare	Erika
Xavier	Bertha	Erika	Clare	Diane	Amy
Yancey	Amy	Diane	Clare	Bertha	Erika
Zeus	Bertha	Diane	Amy	Erika	Clare

Women's Preference Profile

	0 th	1 st	2 nd	3 rd	4 th
Amy	Zeus	Victor	Wyatt	Yancey	Xavier
Bertha	Xavier	Wyatt	Yancey	Victor	Zeus
Clare	Wyatt	Xavier	Yancey	Zeus	Victor
Diane	Victor	Zeus	Yancey	Xavier	Wyatt
Erika	Yancey	Wyatt	Zeus	Xavier	Victor

Xavier proposes to Bertha.

Men's Preference Profile

	0 th	1 st	2 nd	3 rd	4 th
Victor	Bertha	Amy	Diane	Erika	Clare
Wyatt	Diane	Bertha	Amy	Clare	Erika
Xavier	Bertha	Erika	Clare	Diane	Amy
Yancey	Amy	Diane	Clare	Bertha	Erika
Zeus	Bertha	Diane	Amy	Erika	Clare

Women's Preference Profile

	0 th	1 st	2 nd	3 rd	4 th
Amy	Zeus	Victor	Wyatt	Yancey	Xavier
Bertha	Xavier	Wyatt	Yancey	Victor	Zeus
Clare	Wyatt	Xavier	Yancey	Zeus	Victor
Diane	Victor	Zeus	Yancey	Xavier	Wyatt
Erika	Yancey	Wyatt	Zeus	Xavier	Victor

Xavier proposes to Bertha.
- Bertha dumps Victor and accepts Xavier.

How to efficiently query the ranking of a man in a woman's preference list?

In the following case, Xavier proposes to Bertha. Bertha is matched. Now Bertha should find the rank of Xavier and her current partner Victor, to determine whether to accept or reject Xavier.

Men's Preference Profile

	0 th	1 st	2 nd	3 rd	4 th
Victor	Bertha	Amy	Diane	Erika	Clare
Wyatt	Diane	Bertha	Amy	Clare	Erika
Xavier	Bertha	Erika	Clare	Diane	Amy
Yancey	Amy	Diane	Clare	Bertha	Erika
Zeus	Bertha	Diane	Amy	Erika	Clare

Women's Preference Profile

	0 th	1 st	2 nd	3 rd	4 th
Amy	Zeus	Victor	Wyatt	Yancey	Xavier
Bertha	Xavier	Wyatt	Yancey	Victor	Zeus
Clare	Wyatt	Xavier	Yancey	Zeus	Victor
Diane	Victor	Zeus	Yancey	Xavier	Wyatt
Erika	Yancey	Wyatt	Zeus	Xavier	Victor

Xavier proposes to Bertha.

- ▶ Simple solution: using a loop to find the rank of a man according the man's Appearance No. in the woman's preference list. $O(n)$
- ▶ More efficiently solution:
 - 1、 Maintain a reverse list of a woman's preference list.
Index: man's appearance No. \rightarrow value: man's rank
Actually, we don't need man's rank \rightarrow man's appearance No.
 - 2、 using map to store man's appearance No. \rightarrow value: man's rank

Test

- ▶ Construct Test Data:
 - ▶ Generate random names but do not repetitive: Simple and efficient way: w1,w2, w3 ..or m1,m2, m3 ... and so on.
 - ▶ Prefer Lists: generate 1 to n for priority. Random swap 2 elements. You can also construct some special cases, for example, all men's preference lists are the same.
- ▶ Check Results:
 - ▶ Check the pairs number
 - ▶ Check every man has no repetition and exists in men list.
 - ▶ Check every man's partner has no repetition and exists in women list
 - ▶ Check every pair whether satisfy stable match condition.(no unstable pair)

Unstable pair condition

- ▶ woman x and man y are unstable if:
 - x prefers y to its assigned man.
 - y prefers x to its assigned woman.

Pay Attention

- ▶ Object copying
 - ▶ deep copy
 - ▶ shallow copy