# CS612 Algorithm Design and Analysis

## Lecture 1. Introduction and some representative problems

Dongbo Bu

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

# Algorithm

Algorithm design: the art of programming

# V. Vazirani said:

*Our philosophy on the design and exposition of algorithms is nicely illustrated by the following analogy with an espect of Michelangelos's art:*
*A major part of his effort invloved looking for interesting piceces of stone in the quarry and staring at them for long hours to determine the form thay naturally wanted to take. The chisel work exposed, in a minimalistic manner, this form.*

# V. Vazirani said: continued

*By analogy, we would like to start with a clean, simply stated problem.*
*Most of the algorithm design effort actually goes into understanding the algorithmically relevant combinatorial structure of the problem.*
*The algorithm exploits this structure in a minimalistic manner.....*
*with emphasis on stating the structure offered by the problems, and keeping the algorithms minimalistic.*

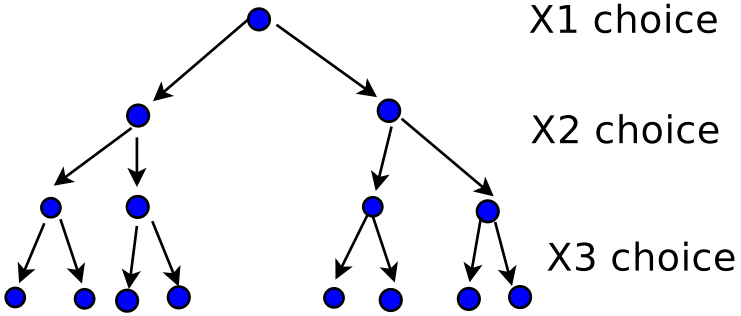
(see two extra slides.)

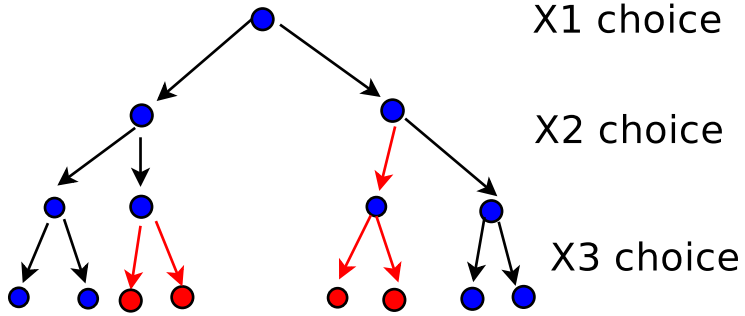
# A first problem: Regular expression matching problem

Contributed by Yanbing Liu, Security lab at ICT.
(see extra slides.)

# A first problem: Regular expression matching problem

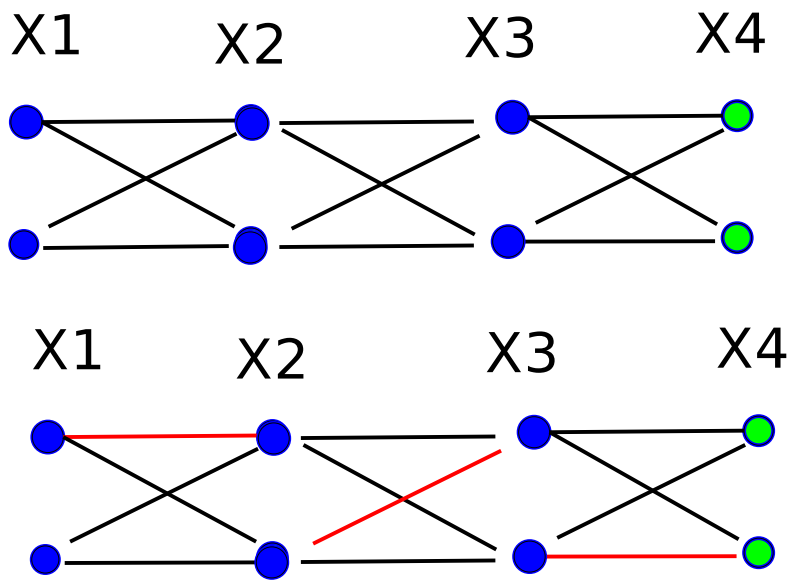
- Key observation: solution=vector;
- Solution space size: $O(2^n)$
- Brute-force: $O(2^n)$



# A first problem: Regular expression matching problem

- Key observation: solution=vector=path; PARTIAL solution
- Key idea: run BFS to check whether the final nodes are reachable; or dynamic programming;
- Time-complexity: $O(4 * n)$



# The second problem: StableMatching problem

In 1962, David Gale and Lloyd Shapley asked a question: Could one design a college admissions process, or a job recruiting process that is self-enforcing?
Motivation: consider some students applying to company for internships.

- Raj accepted an offer from CluNet company;
- WebExodus offers Raj a submmer job later;
- Raj retract his acceptance of the CluNet offer;
- CluNet has to offer a jobs to one of his wait-listed applicants;
- This applicant retracts his acceptance to a company BabelSoft;
- .....

1

[1] Some slides are excerpted from Kevin Wayne's slides.

# Stable Matching – Problem Statement

In mathematics, the *stable marriage problem*(*SMP*) is the problem of finding a stable matching  a matching in which no element of the first matched set prefers an element of the second matched set that also prefers the first element.

## Formalized Definition:

**Input:**

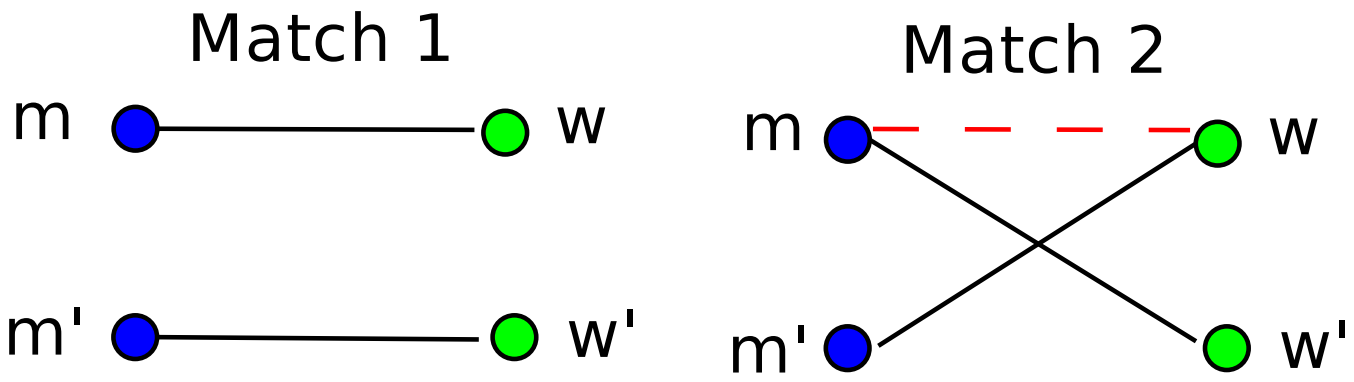
$n$ men and $n$ women, where each person has ranked all members of the opposite sex with a unique number between 1 and $n$ in order of preference.

**Output:**

The matching of the men and women such that there are no unstable match;

# Two examples

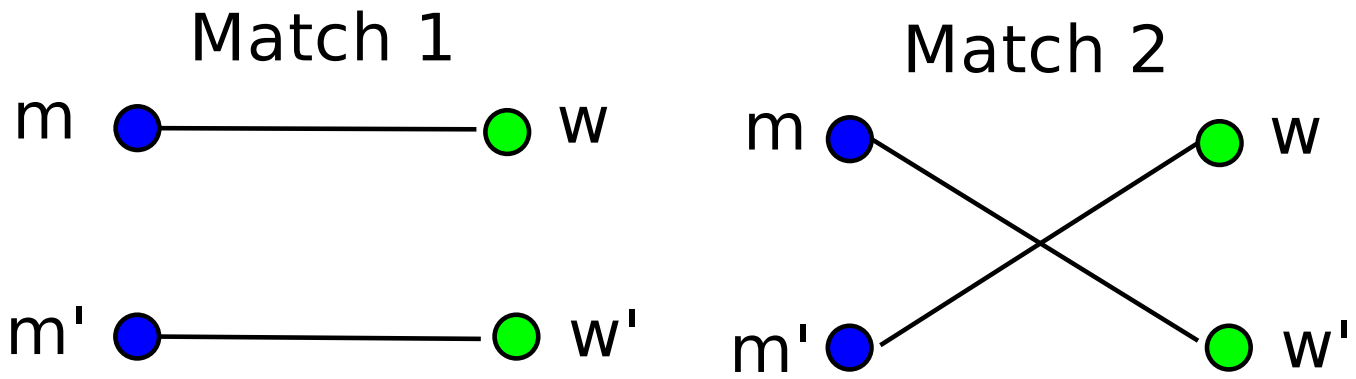
E1: (consensus preference: 1 stable match)
$m$ prefers $w$ to $w'$;
$m'$ prefers $w$ to $w'$;
$w$ prefers $m$ to $m'$;
$w'$ prefers $m$ to $m'$;




**$m$ and $w$ are unstable match:**
— both $m$ and $w$ prefer the other to their current partnets;

# Two examples

E2: (different preference: 2 stable matches)
$m$ prefers $w$ to $w'$;
$m'$ prefers $w'$ to $w$;
$w$ prefers $m'$ to $m$;
$w'$ prefers $m$ to $m'$;



# Stable Matching – Instance

favorite ↓ least favorite ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | Amy | Bertha | Clare |
| Yancey | Bertha | Amy | Clare |
| Zeus | Amy | Bertha | Clare |

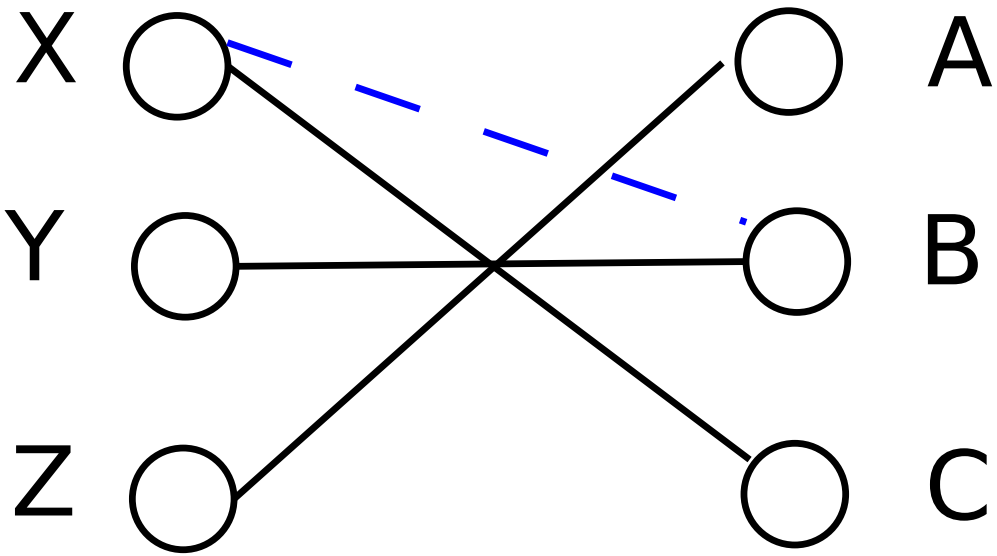
*Men's Preference Profile*

favorite ↓ least favorite ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Amy | Yancey | Xavier | Zeus |
| Bertha | Xavier | Yancey | Zeus |
| Clare | Xavier | Yancey | Zeus |

*Women's Preference Profile*

Is matching $X - C,\ Y - B,\ Z - A$ stable?
No. Bertha and Xavier will hook up.



# Stable Matching – Instance

favorite ↓ least favorite ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | Amy | Bertha | Clare |
| Yancey | Bertha | Amy | Clare |
| Zeus | Amy | Bertha | Clare |

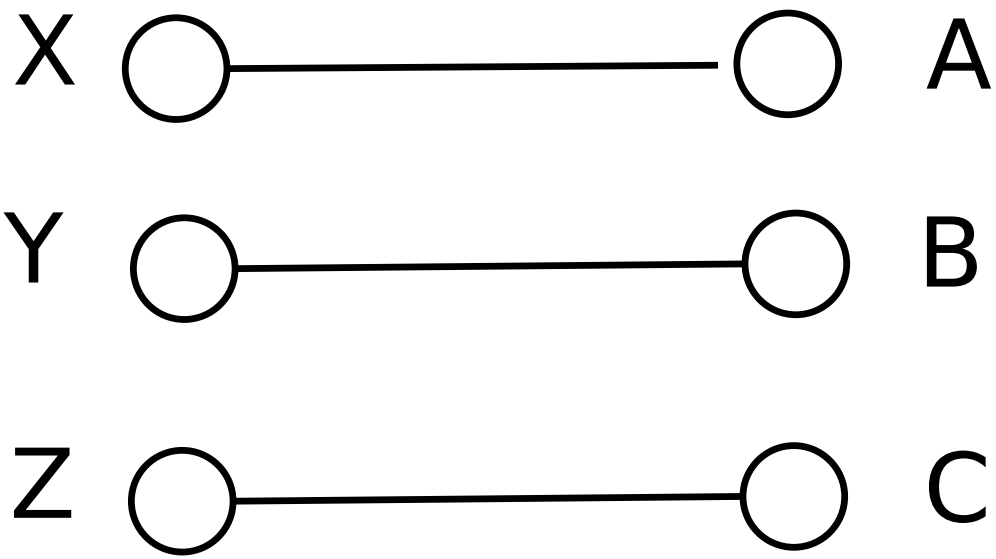
*Men's Preference Profile*

favorite ↓ least favorite ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Amy | Yancey | Xavier | Zeus |
| Bertha | Xavier | Yancey | Zeus |
| Clare | Xavier | Yancey | Zeus |

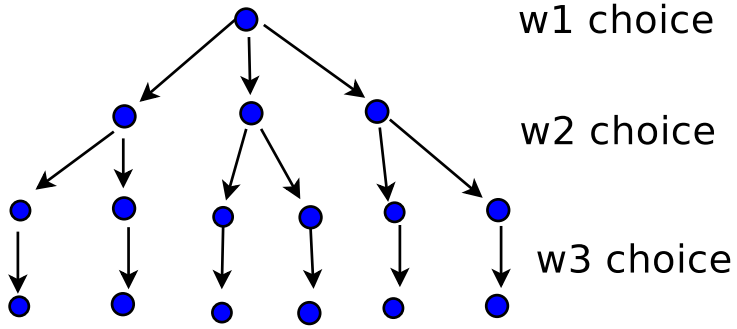
*Women's Preference Profile*

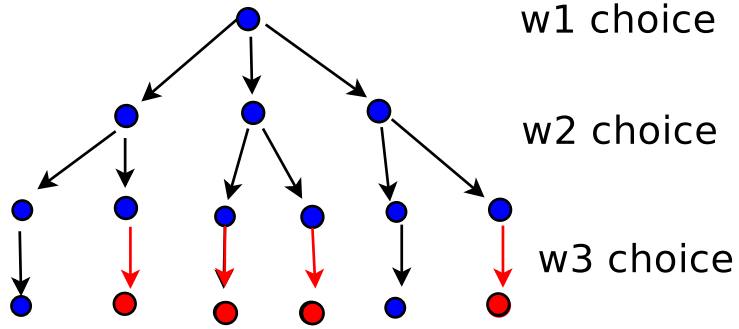
The matching $X - A,\ Y - B,\ Z - C$ is stable.



# Key observations

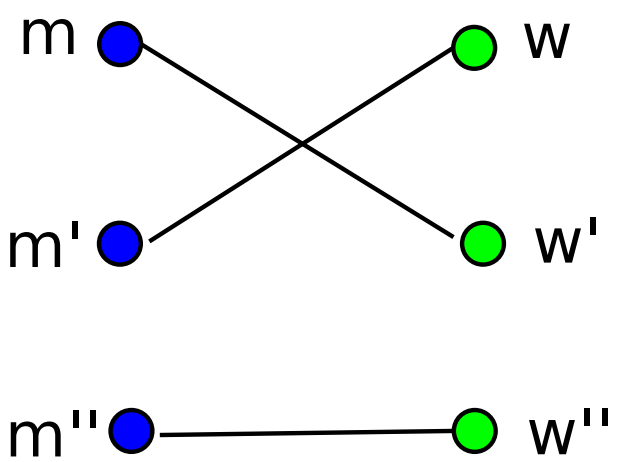
- Solution: match
- Space size: $O(n!)$
- Brute-force: enumerating all possible matches;



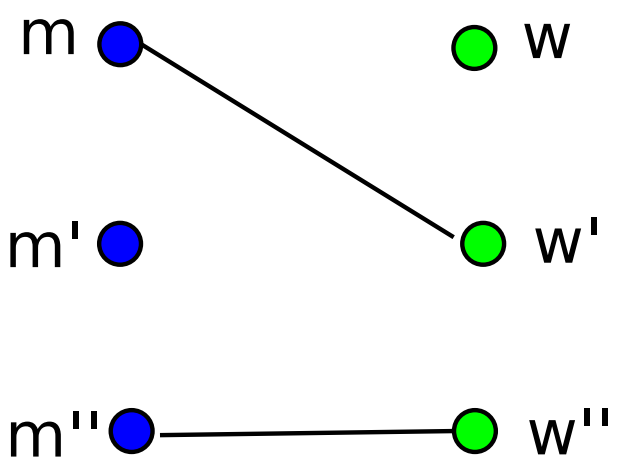



# Key idea

- Solution: match
- Iterative: improving a COMPLETE match;
- Key idea: a "propose-engage" process through maintaining a tentative PARTIAL match;




Complete Solution




Partial Solution

# Stable Matching – G_S algorithm

An intuitive method that guarantees to find a stable matching:

```
for m = 1 to M
     state[m] = FREE;
for w = 1 to W
     state[w] = FREE;
while ( 1 ) {
    if there is no m such that state[m]=FREE
       return;
    select such a m;
    w = 1st woman on m's list to whom m has not yet proposed;
    if( state[w] == free ) {
       state[w] = m;
       state[m] = w;
    } else if( w prefer m to state[w] ){
        state[w] = m;
        state[ state[w] ] = FREE;
    } else {
        ;//simply reject m;
 }
```

(see ppt for demo)

# Key observations of G_S algo

1. Men propose to women in decreasing order of preference.
2. One a woman is matched, she never becomes unmatched.

# Correctness: perfection

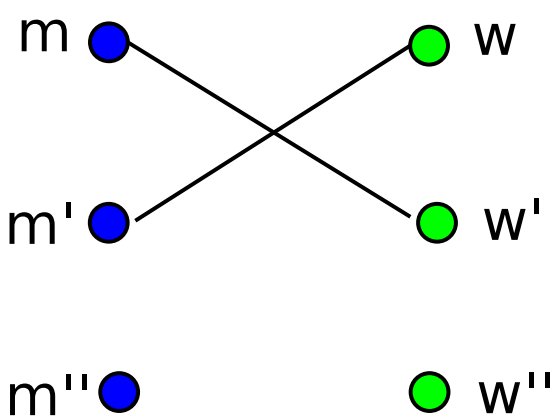
## Theorem

*All men and women get matched.*

## Proof.

Suppose $m''$ is not matched upon termination;

- then there is woman, say $w''$, is not matched;
- then $w''$ should be never proposed to (by Observation 2);
- But $m''$ proposes to everyone. Contradiction.



# Correctness: stability

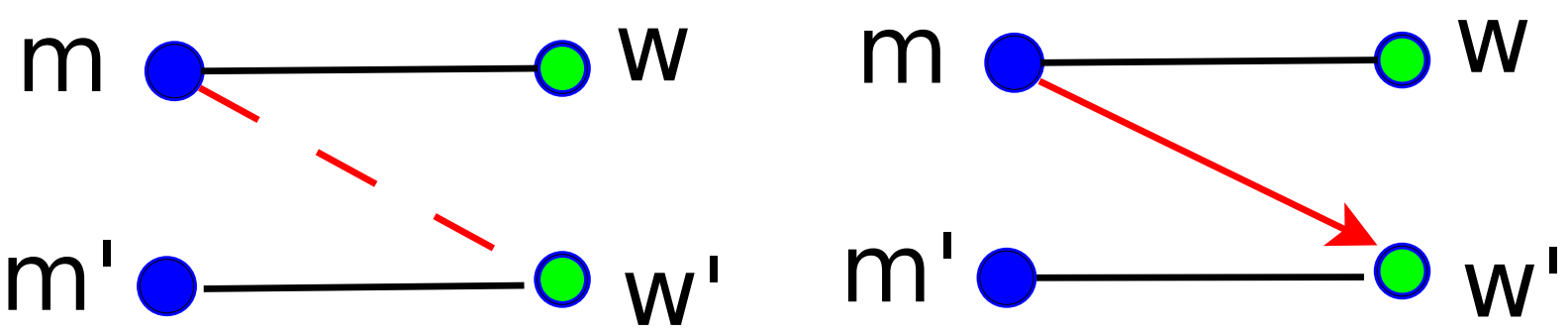
## Theorem

*The finally reported match $S^*$ contains no unstable pairs.*

## Proof.

Suppose $m - w'$ is an unstable pair: each prefers each other to the current partner in $S^*$;

- Case 1: $m$ has proposed to $w'$
  $=>$ $m$ should be rejected by $w'$
  $=>$ $w'$ prefer her GS partner $m'$ to
  $=>$ $m - w'$ is stable.

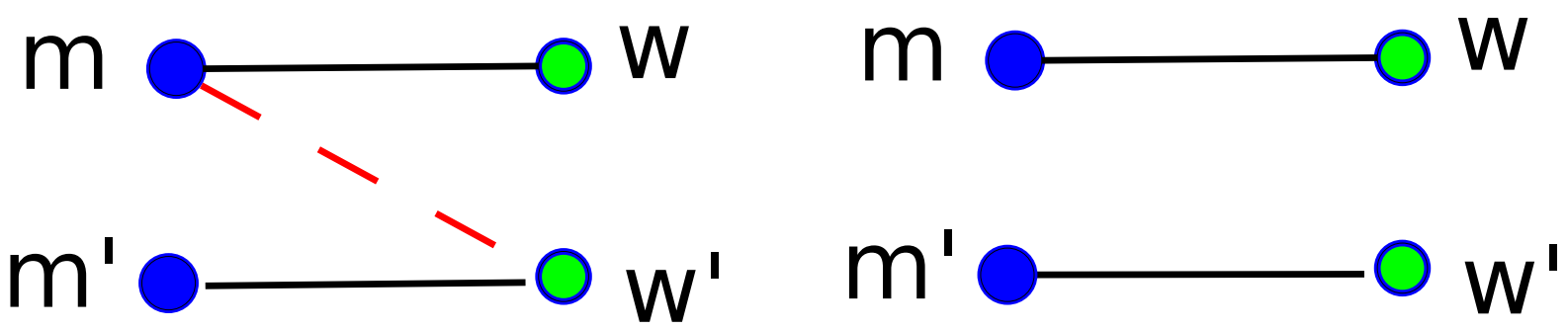


# Correctness: stability

## Theorem

*The finally reported match $S^*$ contains no unstable pairs.*

- Case 2: $m$ never proposed to $w'$
  $=>$ $m$ prefers his GS partner $w$ to $w'$
  $=>$ $m - w'$ is stable;



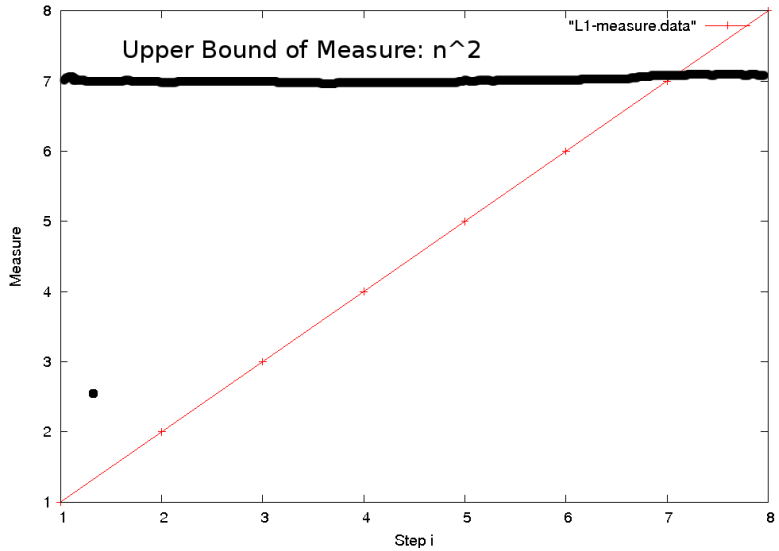
# Analysis: time-complexity I

## Theorem

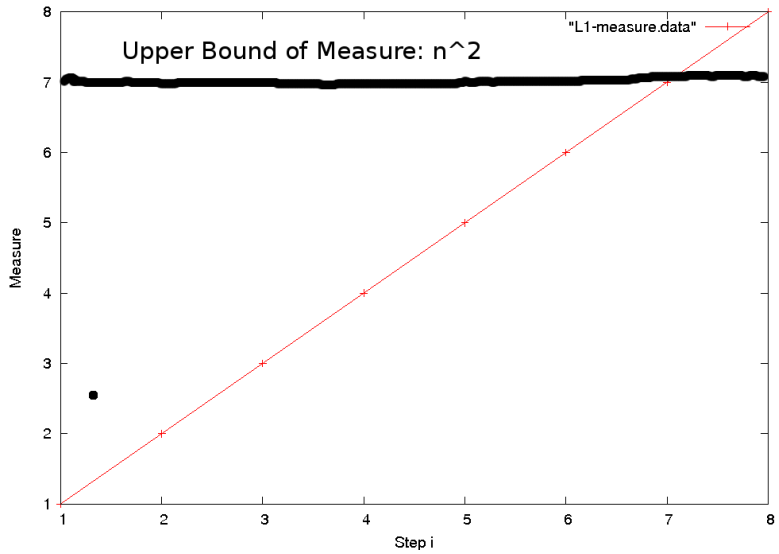
*GS algo ends in $O(n^2)$ steps.*

## Proof.

Key: find a measure of progress for this *while*(1) type loop;
Measure: the number of tried proposals $\#P$;



# Analysis: time-complexity I




Each step: $\#P$ increases at least 1;
Upper bound: $\#P < n^2$
so $T(n) = \#Step < n^2$;
Try other measures:
— the number of matches
— the number of engaged women
— the sum of perference

# A bit strange observation

**Theorem**

*All G_S executions yield the same matching $S^*$.*

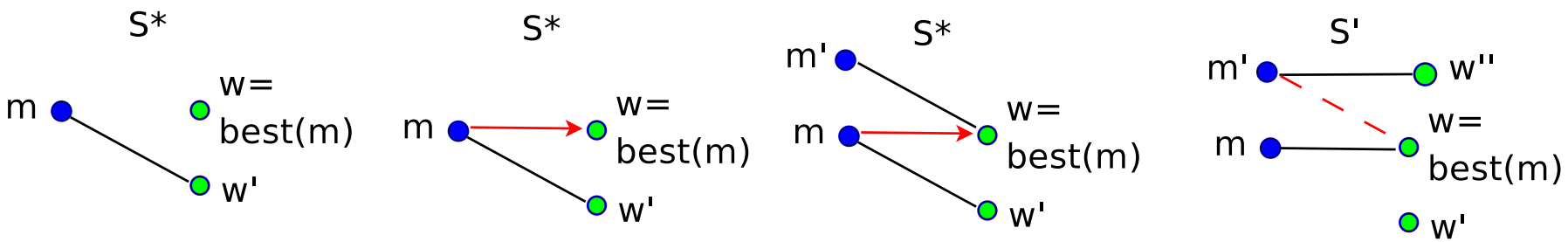
Valid partner: $m$ is a valid partner of $w$ if $m - w$ exists in a stable match;
Man-optimal match: each $m$ receives his best valid partner;
In fact, $S^*$ is man-optimal match.

# A bit strange observation

## Proof.

1. Suppose in $S^*$, there is a match $m - w'$, but $w'$ is not $m$'s best partner; Consider the first such man $m$;
2. then $m$ has already proposed to $w = best(m)$, but was rejected;
3. why? $w = best(m)$ is now paired with a man $m'$ better than $m$;
4. Consider the stable match $S'$ containing $m - w$ as valid partner; $m - w'$ is unstable pair, a contradiction.

□




*Notice: optimal for men, but unfair for women.
it is not obvious that $S^*$ can be reached.