CS711008Z Algorithm Design and Analysis

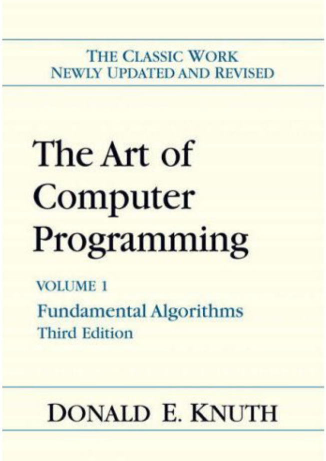
Lecture 1. Introduction and some representative problems [1]

Dongbo Bu

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

[1] The slides are made based on Chapter 1 of Algorithm design. Some slides are excerpted from Kevin Wayne's slides with permission.

Algorithm design: the art of computer programming



V. Vazirani said:

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

V. Vazirani said: $cont'd$

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 minimal manner.....
with emphasis on stating the structure offered by the problems, and keeping the algorithms minimal.

(See two extra slides.)

The first problem: STABLE MATCHING 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 summer 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;
  - ......

Stable Matching – Problem Statement

In mathematics, the Stable Matching problem is the problem of finding a stable matching — a matching in which two agents cannot be found who would prefer each other over their current counterparts.

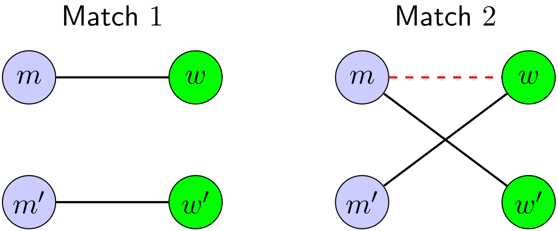
Formalization:

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:
A matching of the men and women such that there is no unstable pair.

Two men and two women: unstable matching

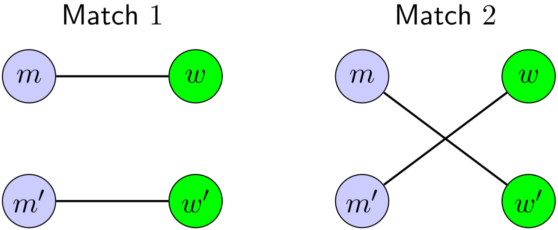
- Example 1: (consensus preference: 1 stable matching)
  m prefers w to w';
  m' prefers w to w';
  w prefers m to m';
  w' prefers m to m';



- In matching 2, m and w form an `unstable pair:` (red, dashed line)
  — both m and w prefer the other to their current partners;

Two men and two women: stable matching

- Example 2: (different preference: 2 stable matchings)
  m prefers w to w';
  m' prefers w' to w;
  w prefers m' to m;
  w' prefers m to m';



- Both matching 1 and 2 are stable.

Three men and three women: unstable matching

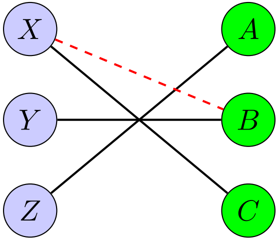
	1st (favorite)	2nd	3rd (least favorite)
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	1st (favorite)	2nd	3rd (least favorite)
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.



Three men and three women: stable matching

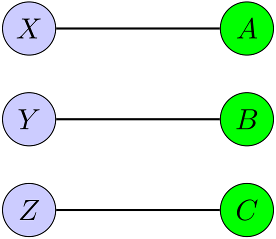
	favorite 1st	2nd	least favorite 3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	favorite 1st	2nd	least favorite 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.



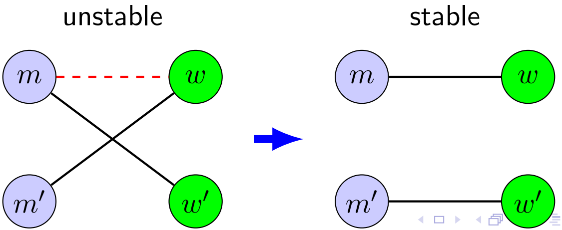
Question: How to find a stable matching?

Trial 1: Improvement strategy

Trial 1: Improvement strategy

- Basic idea: starting from a **complete** matching, and try to **improve** the matching via reducing unstable pairs. If the number of unstable pairs was reduced to 0, then we get a solution.
- SWITCHING operation: making unstable pairs to be stable
- An example of SWITCHING operation:

  m prefers w to w';
  m' prefers w to w';
  w prefers m to m';
  w' prefers m to m';



Trial 1: Improvement strategy

Initializing a matching randomly;
while $\exists$ unstable pairs **do**
    Select an unstable pair $m - w$ arbitrarily ;
    Perform SWITCHING operation to resolve the unstable pair $m - w$;
end while

Improvement strategy: a success case

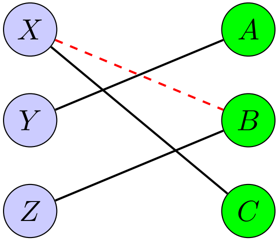
	1st (favorite)	2nd	3rd (least favorite)
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	1st (favorite)	2nd	3rd (least favorite)
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

- Starting from an unstable matching



Improvement strategy: a success case

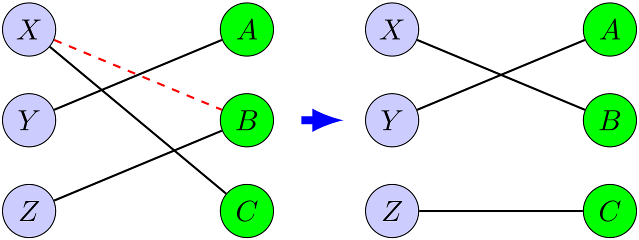
	favorite 1st	2nd	least favorite 3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	favorite 1st	2nd	least favorite 3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

- After one step of switching, we get a stable matching.

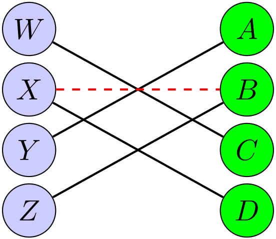


Improvement strategy: a failure case

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

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

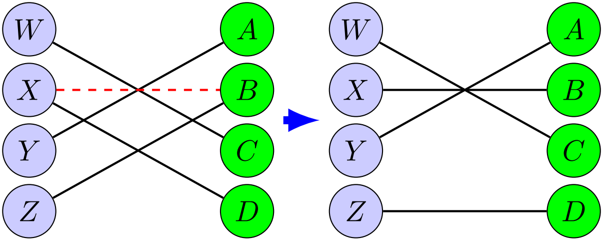
- Starting from an unstable matching



A failure case: Step 1

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

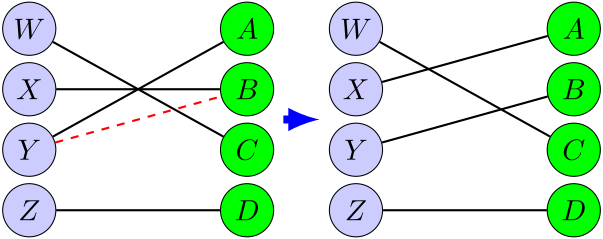
	woman			
	1st	2nd	3rd	4th
A	Z	W	Y	X
B	W	Y	X	Z
C	X	Y	W	Z
D	X	Y	W	Z



A failure case: Step 2

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

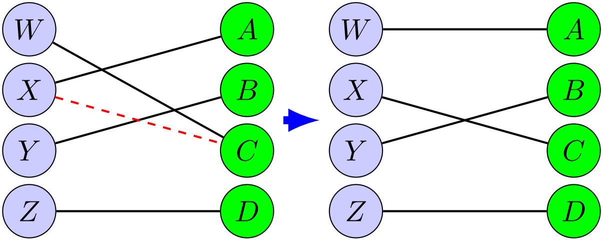
	woman			
	1st	2nd	3rd	4th
A	Z	W	Y	X
B	W	Y	X	Z
C	X	Y	W	Z
D	X	Y	W	Z



A failure case: Step 3

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

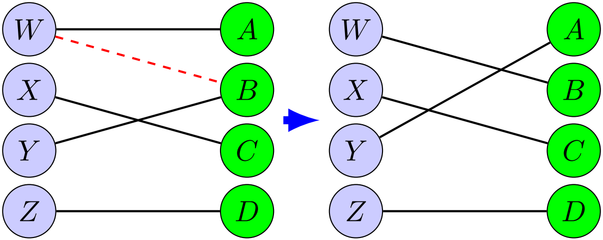
	woman			
	1st	2nd	3rd	4th
A	Z	W	Y	X
B	W	Y	X	Z
C	X	Y	W	Z
D	X	Y	W	Z



A failure case: Step 4

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

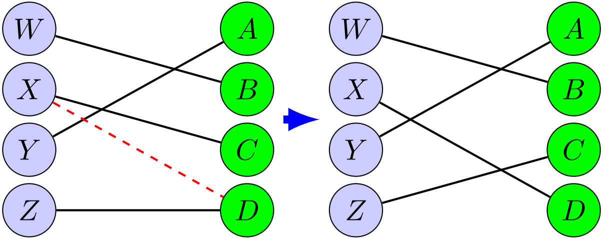
	woman			
	1st	2nd	3rd	4th
A	Z	W	Y	X
B	W	Y	X	Z
C	X	Y	W	Z
D	X	Y	W	Z



A failure case: Step 5

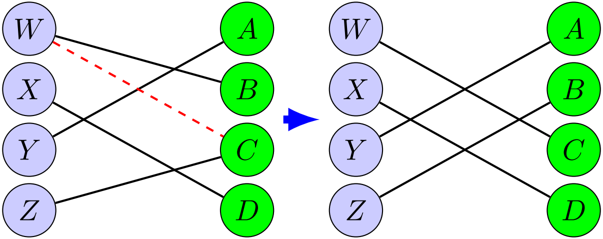
	man			
	1st	2nd	3rd	4th
W	C	B	D	A
X	B	D	C	A
Y	C	D	B	A
Z	D	C	B	A

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



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

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

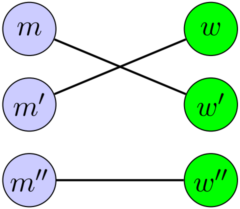


- Failed! Return to the initial matching.

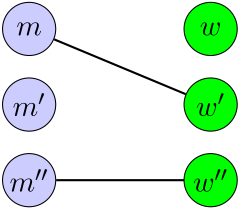
Trial 2: Increment strategy

Trial 2: Increment strategy

- Key observation: the solution is a **compelte** matching.
- Basic idea: Growing up from **partial** matching to **complete** matching, and ensure no unstable pairs during the increment process.
- Implementation: a "propose-engage" process. Man: propose, woman: accept or reject.



complete solution



partial solution

Stable Matching – Gale_Shapley algorithm

```
for m = 1 to M do
    partner[m] = NULL
end for
for w = 1 to W do
    partner[w] = NULL
end for
while TRUE do
    if there is no man m such that partner[m] = NULL then
        return;
    end if
    select such a man m arbitrarily;
    w = the first woman on m's list to whom m have not yet proposed;
    if partner[w] == NULL then
        partner[w] = m; partner[m] = w;
    else if w prefers m to partner[w] then
        partner[partner[w]] = NULL; partner[w] = m; partner[m] = w;
    else
        ; //do nothing means simply rejecting m;
    end if
end while
```

(see ppt for a demo)

Correctness proof

Key observations of Gale_Shapley algorithm

Key observations:

1. Men propose to women in the decreasing order of preference.
2. Once a woman is matched, she never becomes unmatched.
3. When a man proposes, the existing matching might be destroyed.

Correctness: perfection

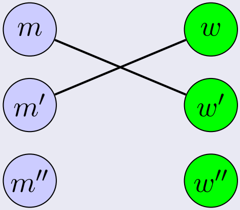
Theorem

All men and women finally 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.



$\square$

Correctness: stability

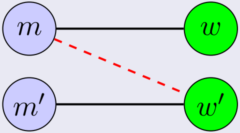
Theorem

At each step of the while loop, the inter-mediate partial match is a stable match. As a special case, the finally reported match S^* contains no unstable pairs.

Proof.

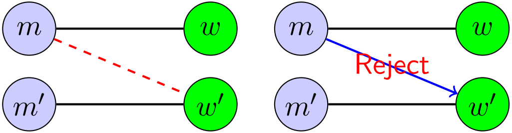
Suppose $m - w'$ is an unstable pair: each prefers the other to the current partner in S^*;

- Case 1: m never proposed to w'
  $\Rightarrow m$ prefers his GS partner w to w'
  $\Rightarrow m - w'$ is stable. A contradiction.



□

- Case 2: m has proposed to w'
  $\Rightarrow$ m should be rejected by w'
  $\Rightarrow$ w' prefer her GS partner m' to m
  $\Rightarrow$ $m - w'$ is stable. A contradiction.



Algorithm analysis: time complexity and space complexity

Analysis: time-complexity

Theorem

Gale-Shapley algorithm 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 $\#P$;

(see an extra slide)

$\square$

Analysis: time-complexity

- Each step: $\#P$ increases at least 1;
- Upper bound: $\#P \leq n^2$
- So $T(n) = \#Step \leq n^2$;
- Try other measures:
  1. the number of matches
  2. the number of engaged women
  3. the sum of preference
- Note: we will revisit Stable Matching problem later.

Time complexity and space complexity

- Time (space) complexity of an algorithm quantifies the time (space) taken by the algorithm.
- Since the time costed by an algorithm grows with the size of the input, it is traditional to describe running time as a function of the input size.
  - **input size**: The best notion of input size depends on the problem being studied.
    - For the STABLE MATCHING problem, the `number of items in the input`, i.e. the number of men, is the natural measure.
    - For the MULTIPLICATION problem, the `total number of bits` needed to represent the input number is the best measure.

Running time: we are interested in its growth rate

- Several simplifications to ease analysis of Gale-Shapley algorithm:
  1. We simply use the number of primitive operations (rather than the exact seconds used) under the assumption that a primitive operation costs constant time. Thus the running time is $T(n) = an^2 + bn + c$ for some constants a, b, c.
  2. We consider only the leading term, i.e. an^2, since the lower order terms are relatively insignificant for large n.
  3. We also ignore the leading term's coefficient a since the it is less significant than the growth rate.
- Thus, we have $T(n) = an^2 + bn + c = O(n^2)$. Here, the letter O denotes order.

Big O notation

- Recall that big O notation is used to describe the `error term` in Taylor series, say:

$$e^x = 1 + x + \frac{x^2}{2} + O(x^3) \text{ as } x \to 0$$

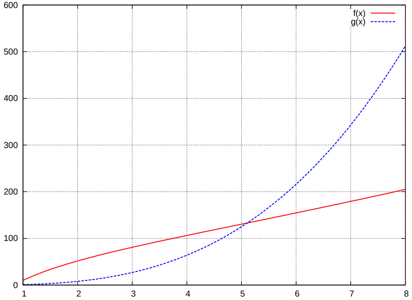


Figure: Example: $f(x) = O(g(x))$ as there exists $c > 0$ (e.g. $c = 1$) and $x_0 = 5$ such that $f(x) < cg(x)$ whenever $x > x_0$

Big Ω and Big Θ notations

- In 1976 D.E. Knuth published a paper to justify his use of the Ω-symbol to describe a stronger property. Knuth wrote: "For all the applications I have seen so far in computer science, a stronger requirement is much more appropriate".
- He defined

$$f(x) = \Omega(g(x)) \Leftrightarrow g(x) = O(f(x))$$

  with the comment: "Although I have changed Hardy and Littlewood's definition of Ω, I feel justified in doing so because their definition is by no means in wide use, and because there are other ways to say what they want to say in the comparatively rare cases when their definition applies".
- Big Θ notation is used to describe "$f(n)$ grows asymptotically as fast as $g(n)$".

$$f(x) = \Theta(g(x)) \Leftrightarrow g(x) = O(f(x)) \text{ and } f(x) = O(g(x)).$$

Extension: a bit strange observation

A bit strange observation

Theorem

Any execution of Gale_Shapley algorithm yields the same matching S^*.

Note:

- The theorem is non-trial since in line 11, an unmatched man m is selected arbitrarily.

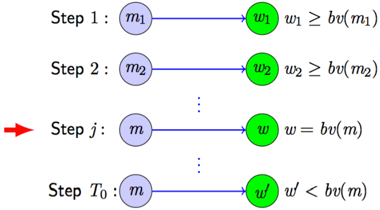
Notations:

- **Valid partner:** w is a valid partner of m if the pair $m - w$ exists in a stable match;
- **Man-optimal match:** each m pairs with his best valid partner, i.e., the best choice he can get.
- In fact, we will prove the theorem by showing that any execution of Gale_Shapley algorithm generates the same **man-optimal match** S^*.

Informally, we say for a man m, $w > w'$ if w is ranked highly than w' in the list of m.

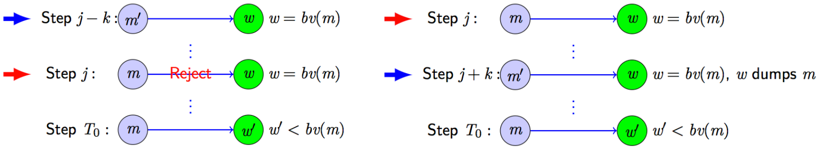
Proof

- A proposal is called ‘‘unlucky’’ if the man proposes to a woman with rank lower than his best valid partner.
- For the sake of contradiction, suppose there is at least one unlucky proposal in an execution.
- Let define $T = \{t|$ at step t, a man proposes to a woman with rank lower than his best valid partner $\}$. Let $T_0 = \min\ T$, i.e. the first unlucky proposal occurs.
- Suppose at time T_0 it is m that proposes to w' such that $w' < best\_valid(m)$. ‘
- Thus before step T_0, m should have proposed to his best valid partner (denoted as w) since w is ranked more highly than w'.



Proof cont'd

- But m finally didn't pair with w. Why? There are two cases:
  1. m was rejected by w directly: w has already paired with m' and in her rank list, m' is better than m (see left-hand panel).
  2. m was accepted by w but was dumped by w afterwards: m' is proposing w and and in her rank list, m' is better than m (see right-hand panel).

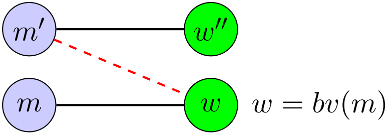


- In both cases, the following property holds:
  1. For w: w prefers m' to m.
  2. For m': $w \geq best\_valid(m')$ since T_0 is the first time that an "unlucky" proposal occurs.

Proof cont'd

- The fact that w is best valid partner of m means that there exists a stable matching, denoted as S', where m pairs with w. Suppose that m' pairs with w'' in S'.

Stable match S'



- Then $m' - w$ should be an unstable pair. (Why?)
- A contradiction. In other words, unlucky proposals never occur in the "propose-engage" process, and any executions of the algorithm yields the same stable matching.

Applications and awards

Applications

- Assigning new doctors to hospitals;
- Assigning students to schools — Public school systems in New York, Boston, Chicago and Denver use an algorithm based on his work to help assign students to schools.
- Finding kidney donors — *For example, a man needs a kidney, and his wife is willing to donate one of hers but she is not a match. Across the country there is a couple in the same position, and it turns out that the wives are a match for the husbands in the opposite couple. In this simple case, the two couples essentially barter their kidneys: Wife A gives her kidney to Husband B, and Wife B gives her kidney to Husband A. It is rare that two couples will serendipitously match each others kidney donation needs this way, and there are often more pairs of donor-recipients involved. Mr. Roths system helps find the most efficient exchange of organs so that the most patients can be saved with the fewest number of pairs involved in a given trade.*

Nobel prize 2012 in Economic Sciences

Figure: L. Shapley (left) and A. Roth (right)

- This year's Prize concerns a central economic problem: how to match different agents as well as possible. For example, students have to be matched with schools, and donors of human organs with patients in need of a transplant. How can such matching be accomplished as efficiently as possible? What methods are beneficial to what groups?
- The prize rewards two scholars who have answered these questions on a journey from abstract theory on stable allocations to practical design of market institutions. The laureates breakthroughs involve figuring out how to properly assign people and things to stable matches when prices are not available to help buyers and sellers pair up.

Nobel prize winner: Lloyd Shapley

- **Lloyd Shapley** used so-called cooperative game theory to study and compare different matching methods. A key issue is to ensure that a matching is stable in the sense that two agents cannot be found who would prefer each other over their current counterparts. Shapley and his colleagues derived specific methods in particular, the so-called Gale-Shapley algorithm that always ensure a stable matching. These methods also limit agents' motives for manipulating the matching process. Shapley was able to show how the specific design of a method may systematically benefit one or the other side of the market.

- **Alvin Roth** recognized that Shapley's theoretical results could clarify the functioning of important markets in practice. In a series of empirical studies, Roth and his colleagues demonstrated that stability is the key to understanding the success of particular market institutions. Roth was later able to substantiate this conclusion in systematic laboratory experiments. He also helped redesign existing institutions for matching new doctors with hospitals, students with schools, and organ donors with patients. These reforms are all based on the Gale-Shapley algorithm, along with modifications that take into account specific circumstances and ethical restrictions, such as the preclusion of side payments.

From theory to practice

- Even though these two researchers worked independently of one another, the combination of Shapley's basic theory and Roth's empirical investigations, experiments and practical design has generated a flourishing field of research and improved the performance of many markets. This year's prize is awarded for an outstanding example of economic engineering.

2

[2] These slides are excerpted from Nobleprize.org