GREEDY ALGORITHMS PT 3

Lecture 13, CMSC 142

Previous Topic(s)

- Fractional Knapsack
- Huffman Encoding

Today's Topics

- Stable Marriage
- Set Cover

Stable Marriage

"when you think all hopes are lost, think again"

- F.R.

Gale-Shapley Algorithm

- Marry men and women in pairs

Gale-Shapley Algorithm

Input: $M = \{m_1, m_2, m_3, \ldots\}$

$W = \{w_1, w_2, w_3, \ldots\}$

Output: $\text{Pairs} = \{(m_1, w_1), (m_2, w_3), (m_3, w_2), \ldots.\}$

How does it work?

- All individuals have ranked members of the opposite set in order of preference
- One of the two sets is chosen to make proposals
- One individual from the proposing group who is not already engaged will propose to their most preferable option who has not already rejected them.
- The person being proposed will:
  - Accept if this is their first offer
  - Reject if this is worse than their current offer
  - Accept if this is better than their current offer.

Men

Freeman	Abby	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	Joy	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Gale-Shapley Algorithm

```
assign each person to be free;
while some man m is free do
begin
        w:= first woman on the m's list to whom m has not proposed;
        if w is free then assign m and w to be engaged {to each other}
        else
                if w prefers m to be her fiance m' then
                        assign m and w to be engaged and m' to be free
                else
                        w rejects m {and m remains free}
end;
output the stable matching consisting of the n engaged pairs
```

Men

Freeman	Abby	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	Joy	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Men

Freeman	Abby	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	Joy	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Abby is not yet engaged to anyone, so Abby accepts Freeman.

Men

Freeman	Abby	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	Joy	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Kira is not yet engaged to anyone, so Kira accepts Edman.

Men

Freeman	Abby	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	Joy	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Joy is not yet engaged to anyone, so Joy accepts Muncy.

Men

Freeman	Abby	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	Joy	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Oh no! Abby is already engaged with Freeman.

But Abby likes Ohtani better than Freeman.

Men

Freeman	Abby	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	Joy	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Oh no! Abby is already engaged with Freeman.

But Abby likes Ohtani better than Freeman.

So Abby accepts Ohtani. Freeman is back to being a lonely guy.

Men

Freeman	~~Abby~~	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	Joy	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Freeman is not one to give up. He checks if Joy is available.

Joy is already engaged to Muncy. But, Joy likes Freeman better.

Men

Freeman	~~Abby~~	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	~~Joy~~	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Freeman is not one to give up. He checks if Joy is available.

Joy is already engaged to Muncy.   But, Joy likes Freeman better.

Men

Freeman	~~Abby~~	Joy	Carla	Kira
Edman	Kira	Joy	Abby	Carla
Muncy	~~Joy~~	Carla	Abby	Kira
Ohtani	Abby	Carla	Joy	Kira

Women

Joy	Freeman	Muncy	Edman	Ohtani
Abby	Muncy	Ohtani	Freeman	Edman
Carla	Ohtani	Edman	Muncy	Freeman
Kira	Muncy	Edman	Freeman	Ohtani

Muncy checks if Carla is still available.

Since Carla is not engaged to anyone, Carla accepts the proposal.

Analysis

- $O(n^2)$

Conclusion

- All should be stable
- Look for the most preferred one.
- Don't worry you will have a partner, at the very least there will.

Set Cover

Introduction

- Suppose I was elected as the new mayor of a town in a remote province.

- The first problem I encounter during my term is Education.

Introduction

- My town doesn't have schools yet! (People in the town are uneducated = probably the reason why I was elected in the first place)

- So, I asked the town cartographer for a map of the town because I want to build schools

Introduction

- I want each school to be located in a barangay, and no student should have to travel more than 5 kilometers to reach one school.

- I also want to build the minimum number of schools needed, as I am constrained by the town budget.

Introduction

- The problem now is where to put the schools so that all children from any barangay can have access to **at least one school**, given the map of the town with the respective distances of each barangay

Set Cover

- **Input**: A collection of subsets $S = \{S_1,..,S_m\}$ of the universal set $U = \{1,..,n\}$

- **Output**: Smallest subset T of S whose union equals the universal set

$$U = \{ 1, 2, 3, 4, 5, 6, 7 \}$$

$$S_a = \{ 3, 7 \} \qquad S_b = \{ 2, 4 \}$$

$$S_c = \{ 3, 4, 5, 6 \} \qquad S_d = \{ 5 \}$$

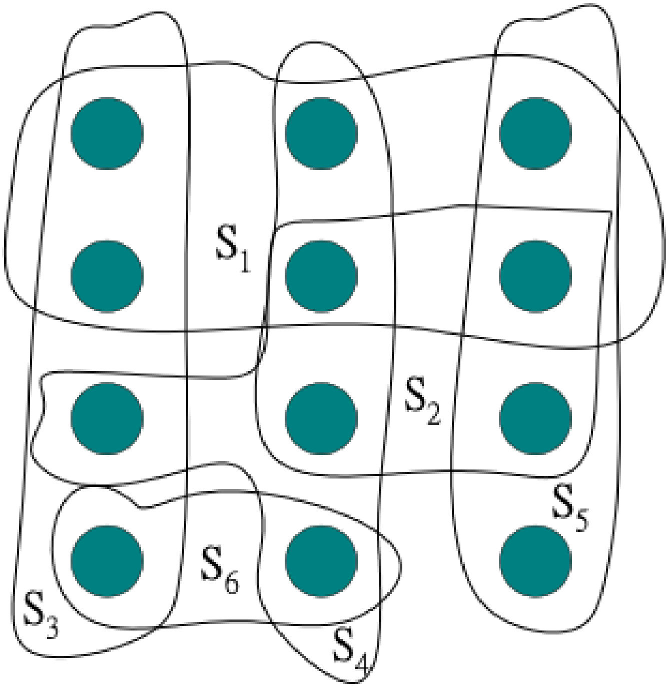
$$S_e = \{ 1 \} \qquad S_f = \{ 1, 2, 6, 7 \}$$

$$k = 2$$

Set Cover

- **Input**: A collection of subsets $S = \{S_1,..,S_m\}$ of the universal set $U = \{1,..,n\}$

- **Output**: Smallest subset T of S whose union equals the universal set



Set Cover

- What is the optimal solution?
- Their union contains all points

Important

- It is important to know if you are allowed to *cover elements more than once.*
- If yes, then this is the Set Cover problem
- Otherwise, it is the Set Packing problem

Set Packing

- It is the same as Set Cover, only it doesn't allow one element to be covered more than once.

Set Cover

- A combinatorial optimization problem (Finding an optimal object from a finite set of objects)

- The study of this problem has led to development of fundamental techniques for the entire field of **approximation algorithms**

Set Cover

- Since this is an **NP-Hard** problem, we can only approximate a solution using the greedy approach

- This is an example of a problem where Greedy algorithm can not solve the optimal solution

- It's not Greedy's fault really, the problem itself is NP-Hard ("It's not you,it's me")

Greedy Algorithm

- We will start with a really lazy and trivial greedy heuristic
- Then, we will continuously improve our greedy heuristic

First Idea

- Take sets by **input order** until all items are covered
- It is a *feasible solution;* eventually, it will cover the whole set
- Gives an upper bound on the objective, or a starting point to improve upon

First Idea

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

First Idea

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

First Idea

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

First Idea

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X
	X			X					

First Idea

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X
	X	X	X	X		X	X		X

First Idea

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X
	X	X	X	X		X	X		X

First Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						
S_2	X	X	X			X	X		X	
S_3		X	X			X	X		X	
S_4							X	X	X	
S_5	X			X	X	X	X			
S_6		X	X					X	X	
	X	X	X	X		X	X	X	X	

First Idea

5	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X
	X	X	X	X	X	X	X	X	X

First Idea

- We got a subset of size 5
- It's better than 6 at least
- **Worst-Case**:

First Idea

- We got a subset of size 5
- It's better than 6 at least
- **Worst-Case**: n

Do you have a better idea?

Second Idea

- Take sets with most elements first

- *Reason:* lots of elements will be covered quicker

Second Idea

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Second Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						2
S_2	X	X	X			X	X		X	6
S_3		X	X			X	X		X	5
S_4							X	X	X	3
S_5	X			X	X	X	X			5
S_6		X	X					X	X	4

Second Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						2
S_2	X	X	X			X	X		X	6
S_3		X	X			X	X		X	5
S_4							X	X	X	3
S_5	X			X	X	X	X			5
S_6		X	X					X	X	4
	X	X	X			X	X		X	

Second Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						2
S_2	X	X	X			X	X		X	6
S_3		X	X			X	X		X	5
S_4							X	X	X	3
S_5	X			X	X	X	X			5
S_6		X	X					X	X	4
	X	X	X			X	X		X	

Second Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						2
S_2	X	X	X			X	X		X	6
S_3		X	X			X	X		X	5
S_4							X	X	X	3
S_5	X			X	X	X	X			5
S_6		X	X					X	X	4
	X	X	X	X	X	X	X		X	

Second Idea

4	1	2	3	4	5	6	7	8	9	
S_1	X			X						2
S_2	X	X	X			X	X		X	6
S_3		X	X			X	X		X	5
S_4							X	X	X	3
S_5	X			X	X	X	X			5
S_6		X	X					X	X	4
	X	X	X	X	X	X	X	X	X	

Second Idea

- Definitely better than first idea
- Can we still improve it and make it smarter?

Third Idea

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Third Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						2
S_2	X	X	X			X	X		X	6
S_3		X	X			X	X		X	5
S_4							X	X	X	3
S_5	X			X	X	X	X			5
S_6		X	X					X	X	4

Third Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						2
S_2	X	X	X			X	X		X	6
S_3		X	X			X	X		X	5
S_4							X	X	X	3
S_5	X			X	X	X	X			5
S_6		X	X					X	X	4
	X	X	X			X	X		X	

Third Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						1
S_2	X	X	X			X	X		X	0
S_3		X	X			X	X		X	0
S_4							X	X	X	1
S_5	X			X	X	X	X			2
S_6		X	X					X	X	1
	X	X	X			X	X		X	

Third Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						1
S_2	X	X	X			X	X		X	0
S_3		X	X			X	X		X	0
S_4							X	X	X	1
S_5	X			X	X	X	X			2
S_6		X	X					X	X	1
	X	X	X	X	X	X	X		X	

Third Idea

	1	2	3	4	5	6	7	8	9	
S_1	X			X						0
S_2	X	X	X			X	X		X	0
S_3		X	X			X	X		X	0
S_4							X	X	X	1
S_5	X			X	X	X	X			0
S_6		X	X					X	X	1
	X	X	X	X	X	X	X		X	

Third Idea

3	1	2	3	4	5	6	7	8	9	
S_1	X			X						0
S_2	X	X	X			X	X		X	0
S_3		X	X			X	X		X	0
S_4							X	X	X	1
S_5	X			X	X	X	X			0
S_6		X	X					X	X	1
	X	X	X	X	X	X	X	X	X	

Third Idea

- Better than the first two heuristics
- But it is still not optimal

Example

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Optimal Solution

2	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X
	X	X	X	X	X	X	X	X	X

Best Greedy Heuristic

- The third idea is the best greedy heuristic we can have for the set cover problem

- *Idea:* At each stage, pick the subset that covers the greatest number of elements not yet covered.

Algorithm

```
GREEDY-SET-COVER(U,S)
    E = U
    T = {}
    while E is not empty
        select S_x ∈ S that maximizes |S_x ∩ E|
        E = E - S_x
        T = T U {S_x}
    return T
```

Improvements

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Improvements

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Improvements

S_2 dominates S_3

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_3		X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Improvements

S_2 dominates S_3

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Improvements

S_2 dominates S_3

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Improvements

S_2 dominates S_3

S_5 is required

	1	2	3	4	5	6	7	8	9
S_1	X			X					
S_2	X	X	X			X	X		X
[illegible]									
S_4							X	X	X
S_5	X			X	X	X	X		
S_6		X	X					X	X

Example

- We can remove S_3, since it is dominated by S_2
- Start with S_5 included
- These kinds of deductions are the essence of **Constraint Programming** (another optimization strategy)

Example

- Using the greedy algorithm, we will select 3 subsets
- But, the optimal solution only has two subsets. (S_2 and S_5)

Set Cover

- Greedy approach gives a good **approximate** solution to the Set Cover
- *Tradeoff:* Exact optimal solution vs. Speed

End of Lecture