



- Stable matching problem. Given n men and n women, and their preferences, find a stable matching if one exists.
- Gale-Shapley algorithm. Guarantees to find a stable matching for any problem instance.
- Q. How to implement GS algorithm efficiently?
- Q. If there are multiple stable matchings, which one does GS find?



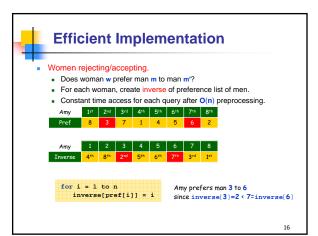
- Problem size
 - N=2n² words
 - 2n people each with a preference list of length n
 - 2n²log n bits
 - specifying an ordering for each preference list takes nlog n bits
- Brute force algorithm
 - Try all n! possible matchings
 - Do any of them work?
- Gale-Shapley Algorithm
 - n² iterations, each costing constant time as follows:



Efficient Implementation

- Efficient implementation. We describe $O(n^2)$ time
- Representing men and women.

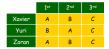
 - Assume men are named 1, ..., n.
 Assume women are named 1', ..., n'.
- - Maintain a list of free men, e.g., in a queue.
 Maintain two arrays wife[m], and husband[w].
 - - set entry to 0 if unmatched
 if m matched to w then wife[m]=w and husband[w]=m
- - For each man, maintain a list of women, ordered by preference.
 - Maintain an array count[m] that counts the number of proposals made by man m.

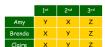




Understanding the Solution

 Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?





- An instance with two stable matchings.
 - A-X, B-Y, C-Z.
 - A-Y, B-X, C-Z.

17

Understanding the Solution

- Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?
- Man-optimal assignment. Each man receives **best** valid partner (according to his preferences).
- Claim. All executions of GS yield a man-optimal assignment, which is a stable matching!
 - No reason a priori to believe that man-optimal assignment is perfect, let alone stable.
 Simultaneously best for each and every man.

18

