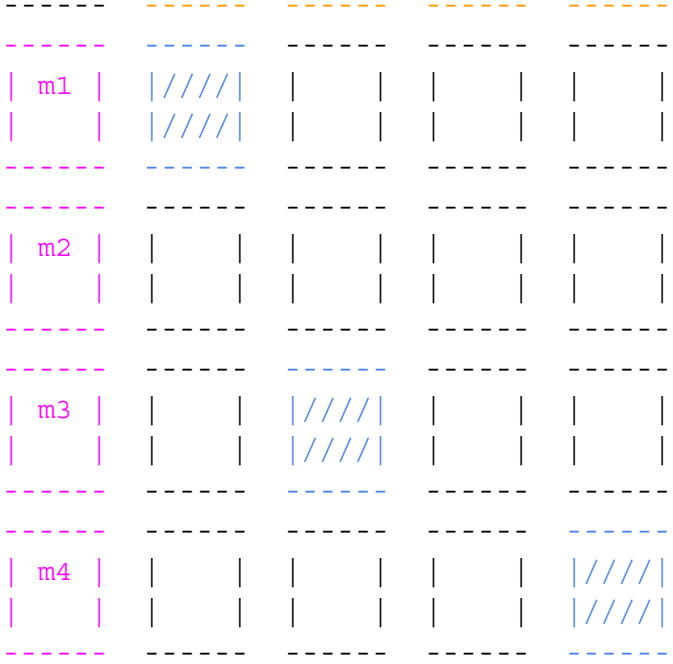
```
PS C:\~> py .\Stable_Marriage_Visualization\main.py

Welcome to An Explanation of the Stable Marriage Problem!

This program will briefly describe the Stable Marriage Problem, some of its
applicaitons, then give a step-by-step visualization for solving such problem.

Press any key to continue . . .
--------------------------------------------------------------------------------

Overview:

The Stable Marriage Problem is concerned with pairing N men and N women, each having
ranked all members of the opposite sex by preference. The goal is to form marriages
such that no two individuals would prefer each other over their assigned partners.

Press any key to continue . . .

It is important to note that we aim to "...model a mathematical problem. We will not,
for instance, consider the realities of same-sex marriage, that individuals don't
necessarily identify as male or female, and that women often propose to men,"
(Austin).

Press any key to continue . . .
--------------------------------------------------------------------------------

Applications:

- Medical Student Assignments: Assigning graduating medical students to their first
  hospital appointments based on their preferences and the preferences of the
hospital.

- Content Delivery Networks: Matching users to servers in a large distributed
  internet service to ensure faster response times and optimal performance.

- Resource Allocation: Matching tasks or resources to agents based on their
  preferences or priorities in scheduling and resource allocation problems.

Press any key to continue . . .
--------------------------------------------------------------------------------

We will now demonstrate an example of solving the Stable Marriage Problem using the
most commonly implemented solving method, the Gale Shapley Algorithm.

Press any key to continue . . .

First, we will need to define the preference lists for our demonstration.

Men's Preferences:
```

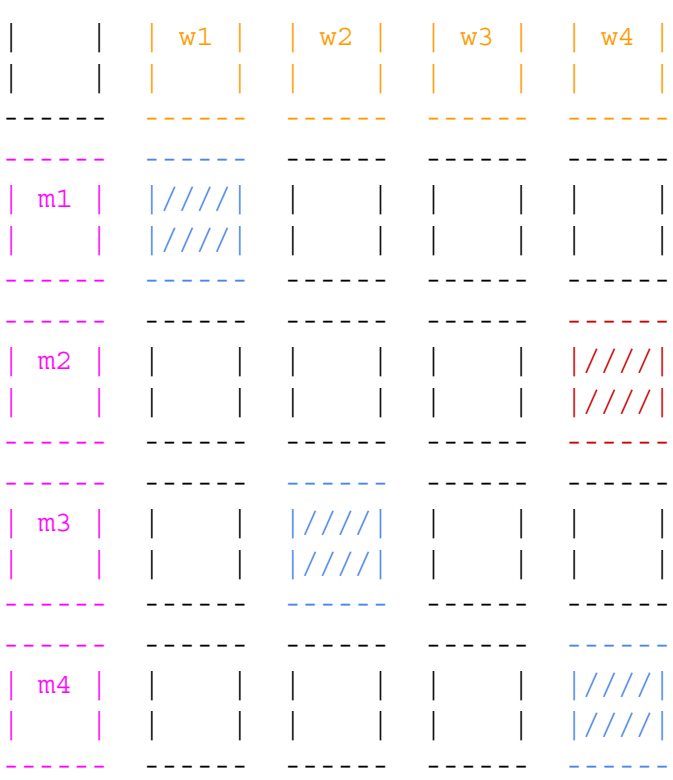
```
        m1: {w1, w2, w3, w4}
        m2: {w1, w4, w3, w2}
        m3: {w2, w1, w3, w4}
        m4: {w4, w2, w3, w1}

Women's Preferences:
        w1: {m4, m3, m1, m2}
        w2: {m2, m4, m1, m3}
        w3: {m4, m1, m2, m3}
        w4: {m3, m2, m1, m4}

Press any key to continue . . .
--------------------------------------------------------------------------------

Step 1A:
------  ------  ------  ------  ------
|    |  | w1 |  | w2 |  | w3 |  | w4 |
|    |  |    |  |    |  |    |  |    |
------  ------  ------  ------  ------
------  ------  ------  ------  ------
| m1 |  |////|  |    |  |    |  |    |
|    |  |////|  |    |  |    |  |    |
------  ------  ------  ------  ------
------  ------  ------  ------  ------
| m2 |  |////|  |    |  |    |  |    |
|    |  |////|  |    |  |    |  |    |
------  ------  ------  ------  ------
------  ------  ------  ------  ------
| m3 |  |    |  |////|  |    |  |    |
|    |  |    |  |////|  |    |  |    |
------  ------  ------  ------  ------
------  ------  ------  ------  ------
| m4 |  |    |  |    |  |    |  |////|
|    |  |    |  |    |  |    |  |////|
------  ------  ------  ------  ------

Each man proposes to the woman he most prefers:
        m1 proposes to w1
        m2 proposes to w1
        m3 proposes to w2
        m4 proposes to w4

Press any key to continue . . .


Step 1B:
------  ------  ------  ------  ------
|    |  | w1 |  | w2 |  | w3 |  | w4 |
|    |  |    |  |    |  |    |  |    |
```



Notice that w1 receives proposals from m1 and m2. She
chooses the proposal from m1 since she prefers m1 to m2.

Press any key to continue . . .

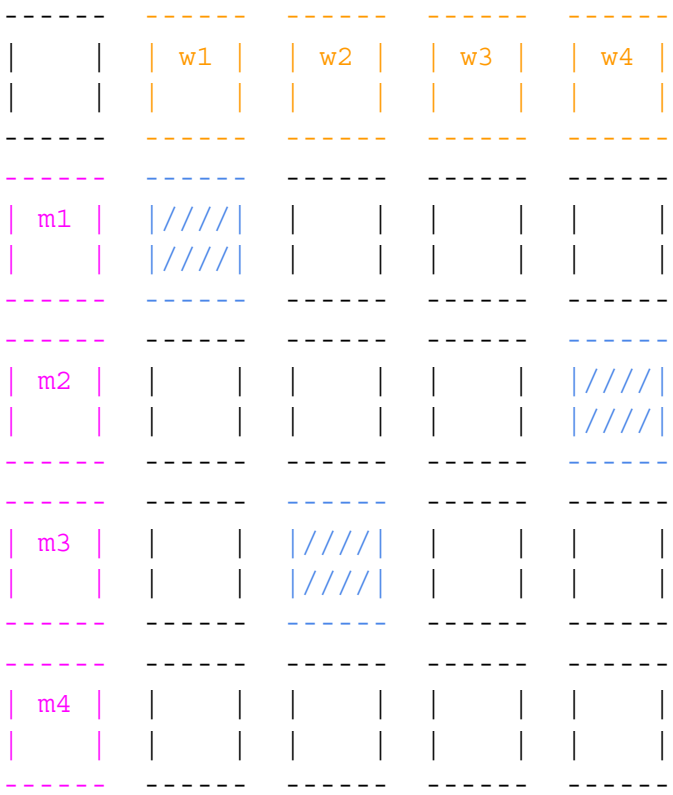
Step 2A:



Since m2 has been rejected by w1, he proposes to his second choice, w4.

Press any key to continue . . .

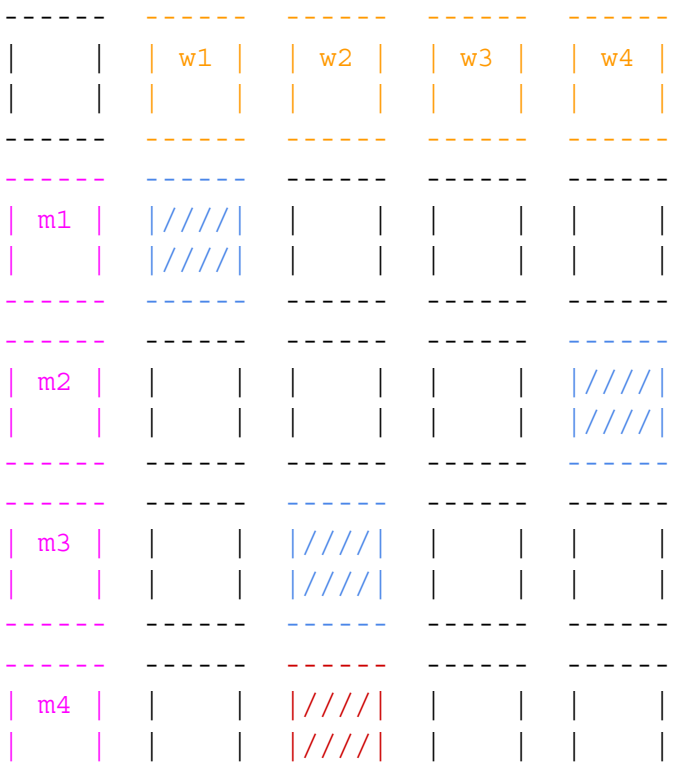
Step 2B:



Now w4 has proposals from m2 and m4 of which she chooses the one from m2.

Press any key to continue . . .

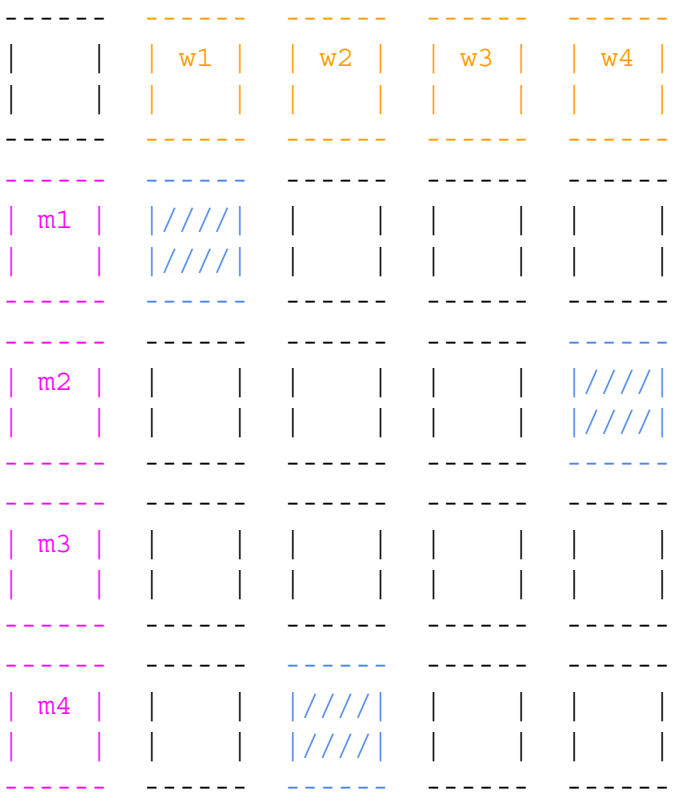
Step 3A:



m4 proposes to w2...

Press any key to continue . . .

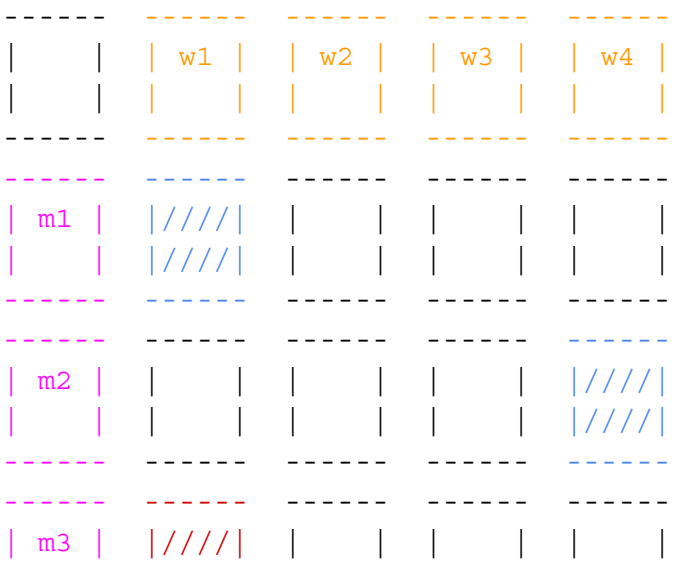
Step 3B:

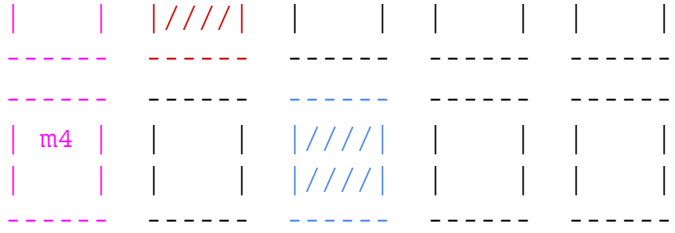


who accepts the proposal and rejects m3.

Press any key to continue . . .

Step 4A:

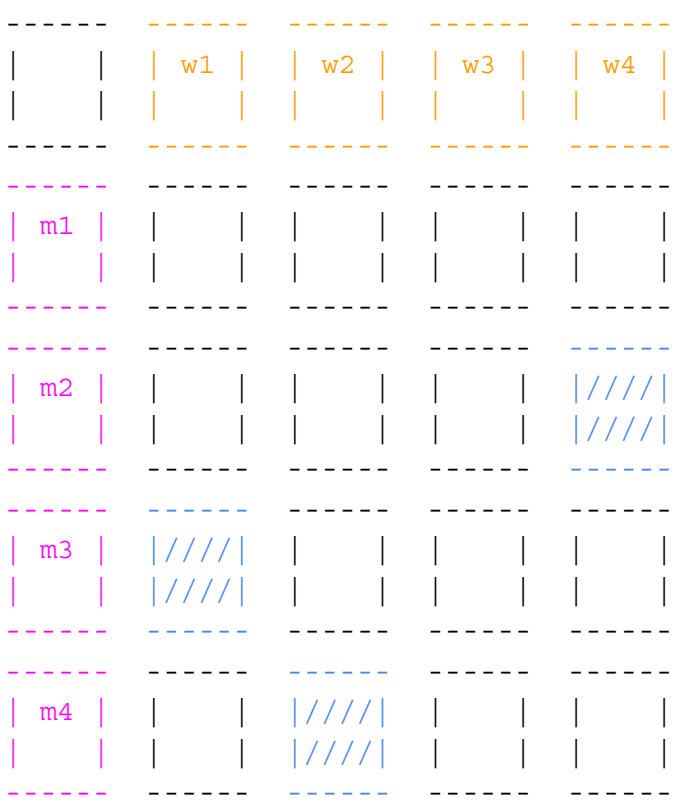




m3 proposes to w1...

Press any key to continue . . .

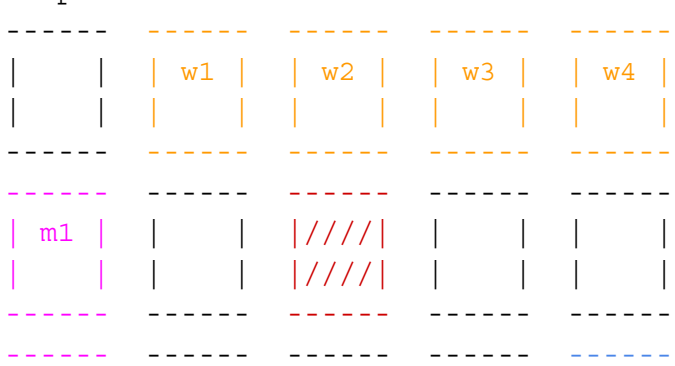
Step 4B:

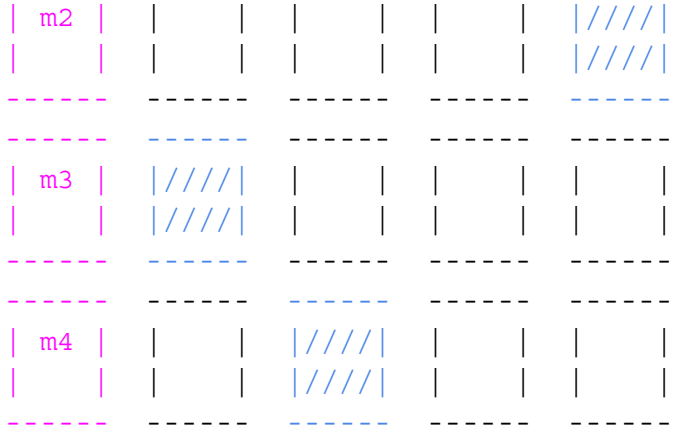


who accepts the proposal and rejects m1.

Press any key to continue . . .

Step 5A:

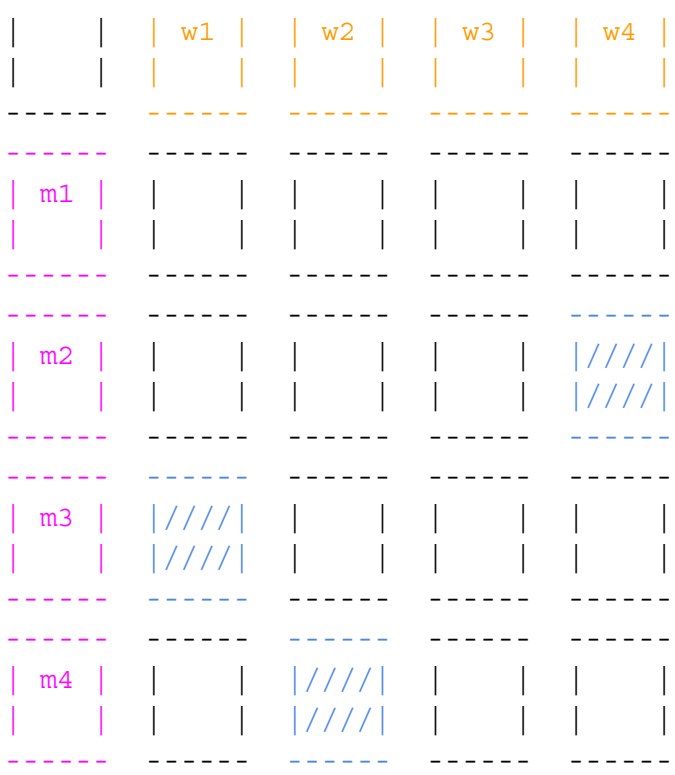




m1 proposes to w2...

Press any key to continue . . .

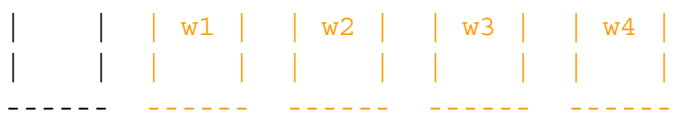
Step 5B:

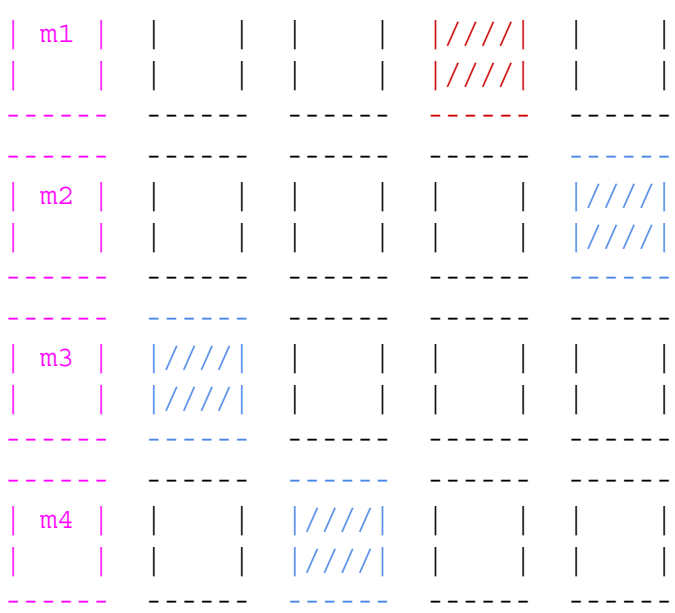


who rejects him because she prefers her current partner, m4.

Press any key to continue . . .

Step 6A:

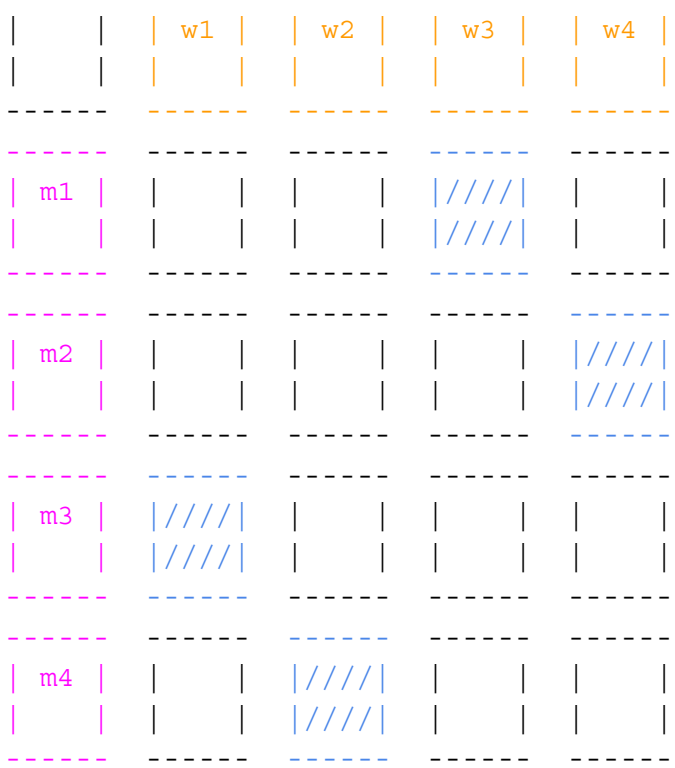




m1 proposes to w3...

Press any key to continue . . .

Step 6B:



who accepts his proposal, which is our final marriage.

Press any key to continue . . .

```
Resulting Marriages:
        m1 is married to w3
        m2 is married to w4
        m3 is married to w1
        m4 is married to w2

As a reminder, our preference lists looked like this:

Men's Preferences:
        m1: {w1, w2, w3, w4}
        m2: {w1, w4, w3, w2}
        m3: {w2, w1, w3, w4}
        m4: {w4, w2, w3, w1}

Women's Preferences:
        w1: {m4, m3, m1, m2}
        w2: {m2, m4, m1, m3}
        w3: {m4, m1, m2, m3}
        w4: {m3, m2, m1, m4}

We can see that the resulting marriages have accomplished the goal of solving the
Stable
Marriage Problem: no two individuals prefer each other over their assigned partners.
--------------------------------------------------------------------------------

Sources:

[1]  Austin, David. "The Stable Marriage Problem and School Choice." American
Mathematical
Society, March 2015, https://www.ams.org/publicoutreach/feature-column/fc-2015-03.
Accessed 20 Nov. 2025.

[2]  Microsoft. Github Copilot, GPT-4.1, OpenAI, 2024. Accessed 20 Nov. 2025; Used to
format graphs and assistance with small debugging tasks.
--------------------------------------------------------------------------------
```