Stable Matching Report

Author 1 and Author 2

April 18, 2024

1 Results

Briefly comment the results, did the script say all your solutions were correct? Approximately how long time does it take for the program to run on the largest input? What takes the majority of the time?

The solution was implemented in a Python script with the Gale-Shapley algorithm that solves the stable marriage problem. The shell script provided validates the solution to have been implemented correctly. The runtimes are summarized in table 1

Table 1: Execution times for different values of N

N	**Time (seconds)**
2	3.8147×10^{-6}
2	4.0531×10^{-6}
4	1.0967×10^{-5}
4	9.0599×10^{-6}
10	9.7752×10^{-6}
100	6.1488×10^{-4}
3000	0.9097
3000	0.6420

2 Implementation details

How did you implement the solution? Which data structures were used? Which modifications to these data structures were used? What is the overall running time? Why?

The Gale-Shapley algorithm was implemented to solve the Stable Marriage Problem. The data structures used include dictionaries for both companies' and students' preferences, a list for the pool of students (‘p‘), and a dictionary for the students who have applied to each company (‘applied‘). The modifications to

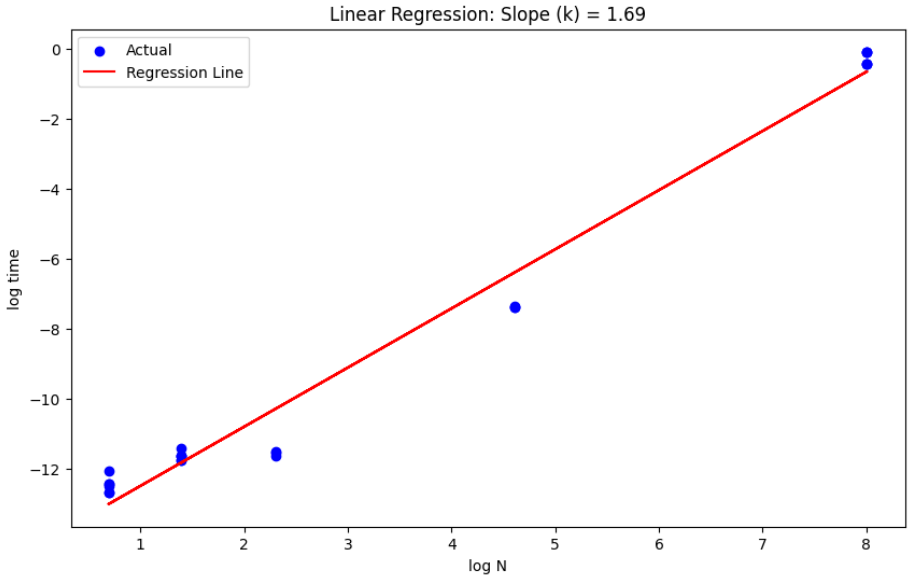

Figure 1: *Linear regression of log plot of running times. Log time (s) against log N which is the size of the dataset in the programming task. The programming task has time complexity $O(n^2)$ which is validated since the slope is below 2.*

the 'applied' data structure occur within the while loop, where students apply to companies based on their preferences. The overall running time of the algorithm is $O(n^2)$ in the worst case, as each student may apply to each company once, and each company must compare the preferences of up to n students. This performance is guaranteed by the fact that each company and student pair is considered at most once.

This problem should have time complexity $O(n^2)$ if implemented correctly, which in practice means that the running time has this is a upper bound in the slope of the curve in the figure **??**

2.1 Applications Gale-Shapley

Medical Residencies: The Gale-Shapley algorithm matches medical students to residency programs, ensuring stable placements where neither student nor hospital would prefer a different match.

School Admissions: It is used to allocate students to schools based on mutual preferences, maximizing satisfaction in large urban districts.

Labor Markets: The algorithm matches workers with firms to ensure stable and mutually preferred employment relationships.

Kidney Exchange Programs: Adapted versions help identify chains of compatible donor-recipient pairs for kidney transplants, optimizing matches and outcomes.

Roommate Assignments: It solves the Roommates Problem by pairing individuals in a stable and mutually agreeable manner.

Online Dating and Social Networks: Principles of the algorithm inform matching systems to enhance stability and user satisfaction on dating platforms.

Task Assignment: The algorithm assigns computing tasks to resources, ensuring stable and efficient resource utilization.