Worksheet 1: SMP

Rin Meng 51940633
Kevin Zhang 10811057
Mika Panagsagan 29679552

January 14, 2025

1 4. Evaluate simple algorithmic approaches, such as brute force.

1. Bound by the time $t(n)$ we can say that our worst-case scenario of our brute force algorithm is $O(n!)$.

   - The number of ways to match n employers and n students is $n!$.
   - For each employer, we must check each student's preference list and for each student, we must check each employer's preference list, and that is $O(n^2)$.
   - This results in a worst-case time complexity of $O(n! \cdot n^2)$.
   - Since $n!$ grows faster than n^2, we can simplify this to $O(n!)$.

2. Each potential solution is a perfect matching between n employers and n students, therefore the total number of potential solutions is $n!$.

3. Overall worst-case running time of the brute force will always be $O(n!)$.

   - We generate $n!$ potential solutions.
   - For each solution, we spend $O(n^2)$ time to check if it is stable.
   - Since generating a solution takes negligible time, compared to checking it, the dominant term is

$$O(n! \cdot n^2) = O(n^2)$$

- Since $n!$ grows faster than n^2, we can simplify this to $O(n!)$.

Therefore, the brute force algorithm has a factorial time complexity of $O(n!)$, which is extremely inefficient for large n.

2 Design a better algorithm.

We can do this by using the Gale-Shapley algorithm.

1. Input
   (a) Two preference lists, one for students and one for employers.
   (b) The number of students and employers, n.
2. Steps:
   (a) Initialization: Create an empty list of matched pairs. All applicants are initially unmatched, and all employers are initially unmatched.
   (b) Proposal Phase: Each unmatched applicant proposes to the first employer on their preference list who has not already rejected them.
   (c) Employer's response:
      i. Each employer receives proposals and considers them:
         - If they are unmatched, they accept the proposal.
         - If they are already matched but prefer the new applicant over their current match, they reject the current match and accept the new proposal.
         - If they prefer their current match, they reject the new proposal.
   (d) Repeat: Applicants who have been rejected by all employers or who haven't yet been matched will propose to the next employer on their list.
   (e) Termination: The algorithm terminates when no applicants are left to propose or when everyone is matched. At this point, we have a stable matching.

3. Time complexity:

   (a) Time per proposal, each student proposes to at most n employers, and each employer receives at most n proposals, so the time per proposal is $O(n)$.

   (b) Total time complexity, since there are n students and n employers, the total time complexity is $O(n^2)$.

4. Walkthrough:

   (a) Let $n = 3$, S be the student set, and E be the employer set, M be the matching set, and P be the preference list.

   (b) $S = \{s_1, s_2, s_3\}$ and $E = \{e_1, e_2, e_3\}$, $M = \emptyset$.

   (c) $P(s_1) = \{e_1, e_2, e_3\}$, $P(s_2) = \{e_2, e_1, e_3\}$, $P(s_3) = \{e_3, e_2, e_1\}$.

   (d) $P(e_1) = \{s_1, s_2, s_3\}$, $P(e_2) = \{s_2, s_3, s_1\}$, $P(e_3) = \{s_3, s_1, s_2\}$.

      i. s_1 proposes to e_1, e_1 accepts.

      ii. s_2 proposes to e_2, e_2 accepts.

      iii. s_3 proposes to e_3, e_3 accepts.

   (e) Terminate: we the most stable matching, which priotiizes the students.

   $$M = \{(s_1, e_1), (s_2, e_2), (s_3, e_3)\}$$

5. Challenges:

   (a) Let $n = 3$, S be the student set, and E be the employer set, M be the matching set, and P be the preference list.

   (b) $S = \{s_1, s_2, s_3\}$ and $E = \{e_1, e_2, e_3\}$, $M = \emptyset$.

   (c) $P(s_1) = \{e_2, e_3\}$, $P(s_2) = \{e_1, e_3\}$, $P(s_3) = \{e_2, e_1\}$.

   (d) $P(e_1) = \{s_1, s_2, s_3\}$, $P(e_2) = \{s_2, s_3, s_1\}$, $P(e_3) = \{s_3, s_1, s_2\}$.

      i. s_1 proposes to e_2, e_2 accepts.

      ii. s_2 proposes to e_1, e_1 accepts.

      iii. s_3 proposes to e_2, e_2 accepts.

   $$M = \{(s_1, e_2), (s_2, e_1), (s_3, e_2)\}$$

iv. Now we have to goto the employer's preference list, since there are two primary choice for student s_1 and s_3, we have to reject one of them.

v. e_1 stays the same, goes with s_2.

vi. e_3 rejects s_1 and accepts s_2.

vii. e_2 rejects s_2 and accepts s_3.

$$M = \{(s_1, e_2), (s_2, e_1), (s_3, e_3)\}$$