Discussion 2A

CS 70, Summer 2024

1 Induction (Continued)

(a) Let F_i be the i^{th} Fibonacci number, defined by $F_{i+2} = F_{i+1} + F_i$ and $F_0 = 0$, $F_1 = 1$. Prove that

$$\sum_{i=0}^{n} F_i^2 = F_n F_{n+1}.$$

(b) Show that any integer $n \geq 1$ can be decomposed into $n = F_{i_1} + F_{i_2} + \ldots + F_{i_k}$ where $F_{i_1}, F_{i_2}, \ldots, F_{i_k}$ are distinct Fibonacci numbers.

2 Stable Matching

Consider the set of jobs $J = \{1, 2, 3\}$ and the set of candidates $C = \{A, B, C\}$ with the following preferences.

Jobs	Candidates
1	A $\succ$ B $\succ$ C
2	B $\succ$ A $\succ$ C
3	A $\succ$ B $\succ$ C

Candidates	Jobs
A	2 $\succ$ 1 $\succ$ 3
B	1 $\succ$ 3 $\succ$ 2
C	1 $\succ$ 2 $\succ$ 3

Run the traditional propose-and-reject algorithm on this example. Determine the number of days the algorithm takes to terminate as well as the stable matching the algorithm constructs.

3 Propose-and-Reject Proofs

Prove the following statements about the traditional propose-and-reject algorithm.

(a) In any execution of the algorithm, if a candidate receives an offer on day i, then they receive some offer on every day thereafter until termination.

(b) In any execution of the algorithm, if a candidate receives no offers on day i, then they receive no offers on any previous day j, $1 \leq j < i$.

(c) In any execution of the algorithm, there is at least one candidate who only receives a single offer.

4 Job Optimality

In this problem, we will walk through the proof that the propose-and-reject algorithm outputs the job-optimal matching.

For any job J, let M_J be the stable matching in which J is matched to its most preferred partner among all stable matchings. Let that partner be $C^*(J)$. We call $C^*(J)$ the *optimal candidate* for J.

(a) Construct an example with two jobs and two candidates where there is a job J such that $C^*(J)$ is not J's most preferred candidate.

(b) We will now prove that no job J will be rejected by its optimal candidate $C^*(J)$ under the propose-and-reject algorithm.

We will use induction, and prove the stronger claim that for each $t \in \mathbb{N}$, by the end of day t of the propose-and-reject algorithm, no job will have been rejected by its optimal candidate. The beginning of the proof is provided.

Base case. On day 0, no rejections have occurred, so no job has been rejected by its optimal candidate.

Induction case.

Induction hypothesis. Suppose that for each day $1, 2, \ldots, t-1$, no job J has ever been rejected by its optimal candidate $C^*(J)$.

Induction step. Consider day t. For the sake of contradiction, suppose that there exists a job J which was rejected by its optimal candidate $C^*(J)$ on day t in favor of an offer from another job $J^* \neq J$.

...

First show that $c^*(j)$ must prefer j^* to j.

(c) Using the induction hypothesis, show that j^* must also prefer $c^*(j)$ to its optimal candidate $c^*(j^*)$.

(d) Prove that j^* and $c^*(j)$ must be a rogue couple in the matching M_j.

(e) Finish the proof that no job will be rejected by its optimal candidate.

Note. There is another proof which, instead of using induction, uses the *well-ordering principle*. See Note 4 and Homework 1 Problem 4 to learn more.