

1 Reverse Preference Stable Matching

A witch bewitches all the jobs just before a run of the propose-and-reject algorithm, so they propose in reverse order of their preference lists (equivalent to the jobs giving reversed lists as inputs to the algorithm). The spell wears off after the algorithm is run and a matching is obtained. **Only the jobs reverse their preference lists, and after the spell wears off, the jobs remember their original preferences.**

(a) Find the output matchings for the following two instances.

(i)	j.	pref. ($\succ$)		c.	pref. ($\succ$)	
	1	A	B	A	2	1
	2	A	B	B	1	2
(ii)	j.	pref. ($\succ$)		c.	pref. ($\succ$)	
	1	A	B	A	1	2
	2	B	A	B	2	1

(b) Prove that for any output of the bewitched propose-and-reject algorithm, the matching is either unstable or all the candidates got their first choice job.

2 Error Correction

Suppose we want to send a message of n packets (viewed as an integer modulo some large prime q) across an unreliable link. We anticipate that up to 7 packets may be corrupted, so we send a codeword consisting of $n + m$ points $P(1), \dots, P(n + m)$ on some $\text{GF}(q)$ polynomial $P(x)$ of degree $n - 1$. Answer the following questions about this scenario.

- (a) What value should we choose for m ? No justification required.

- (b) Suppose that in reality 10 packets are corrupted, and we receive the values $r_1, r_2, \dots, r_{n+m}$, where m is as in part (a). It is still possible to recover the polynomial P if we are told the locations of some of the errors. What is the minimum number of error locations we need to know to achieve this? No justification required.

- (c) Let ℓ be the minimum number of error locations you found in part (b), and suppose we are told that those error locations are at positions $1, 2, \dots, \ell$. Briefly explain how you would recover the original message given this information. Be sure to specify the degrees of the polynomials $E(x)$ and $Q(x)$ and the points on which they should be evaluated. Your answer should be in terms of the variables m and ℓ defined in parts (a) and (b), respectively. You may assume familiarity with the Berlekamp–Welch algorithm, but you should indicate how to apply it in this setting.

3 Anna's Blackboard

Anna writes down an arbitrary positive integer n on a blackboard. She then repeatedly performs the following operation, where k is the number currently on the board:

- if k is a multiple of 3 then replace k by $k/3$; else replace k by $k + 1$

Prove that, for any starting integer n , Anna will always eventually write the number 1 on the board.

4 Intensive Swim Workout

A swimmer swims in a pool where 70 laps is equivalent to 1 mile. (By “lap” here we mean one length of the pool.) She uses the following exhausting workout regime:

- In alternate laps, she puts her fins on and takes them off.
- She cycles through five breathing patterns (labeled 1,2,3,4,5), switching to the next one at the end of each lap.
- She swims seven different strokes (1=freestyle, 2=breaststroke, 3=butterfly, 4=backstroke, 5=front kick, 6=back kick, 7=sidestroke), switching to the next one at the end of each lap.

She always starts her workout with fins off, breathing pattern 1, freestyle.

One day she loses count of how many laps she swam, but she does know she swam more than one mile and less than two miles. She also knows that her last lap was with fins on, breathing pattern 3, and back kick.

(a) Explain clearly why she has enough information to deduce exactly how many laps she swam.

(b) How many laps did she swim? [**Note:** You may find it easier to do this calculation by hand rather than using a standard algorithm.]

5 Graph Proofs

All graphs are simple and undirected unless otherwise specified.

- (a) Consider a simple connected planar graph with e edges and $v > 2$ vertices. Prove that if $e < 3v - 6$, then there is a face of size at least 4 (that is, there exists a face bounded by at least 4 edges). (Recall that in a simple planar graph, every face has size at least 3.)

- (b) Consider a bipartite graph $G = (V, E)$ with an Eulerian tour. Give a method to partition the edges of the graph into two graphs where every vertex has exactly half the degree, and justify your answer.

That is, form $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, where $E_1 \cup E_2 = E$, $E_1 \cap E_2 = \emptyset$, and every vertex v has $d_1(v) = d_2(v) = d(v)/2$ where $d(v)$ is the degree of a vertex in G , and $d_i(v)$ is the degree of v in G_i .

6 Sleepiness

- (a) Consider a set of intervals $[s_1, e_1], \dots, [s_n, e_n]$ where s_i is the start time and e_i is the end time of an interval. The associated interval graph has a vertex for each interval and an edge between any pair of intervals that overlap; for example, if $i_1 = [3, 5]$ and $i_2 = [4, 6]$ and $i_3 = [6, 7]$, we have edges $\{i_1, i_2\}$ and $\{i_2, i_3\}$ but no edge between i_1 and i_3 .

Prove that if there is a cycle in an interval graph then there is a point in time where at least 3 intervals overlap.

- (b) Seven students each fall asleep three times during CS 70 lecture. Furthermore, for every two of these seven students, there exists a time when both of them are sleeping. Prove that there must be some time when at least three students were sleeping at once.