Homework 2

CS 70, Summer 2024

Due by Friday, July 5th at 11:59 PM

1 Stable Matching Proofs

(a) We prove the statement by contradiction. Suppose that the propose-and-reject and the flipped propose-and-reject algorithm output the same stable matching S, and that there exists some other stable matching $S \neq S'$.

Then there must exist some candidate C and two jobs $J \neq J'$ such that $(C, J) \in S$ but $(C, J') \in S'$. There are two possible cases.

(1) C prefers J' to J. This yields a contradiction, since the S is produced by the flipped propose-and-reject algorithm and is therefore candidate-optimal. That is, J is the most preferred job which C could be ranked with in any stable matching.

(i) C prefers J to J'. This yields a contradiction, since S is produced by the propose-and-reject algorithm and is therefore candidate-pessimal. That is, J is the least preferred job which C could be ranked with in any stable matching.

In either case, we get a contradiction. Therefore our assumption that the two stable matchings S and S' were different must have been incorrect.

Therefore we have shown that there is only one stable matching.

(b) We disprove the statement. The propose-and-reject algorithm produces a job-optimal matching, and any job J only proposes to candidates which they like at least as much as the final candidate with which they are matched.

If there exists a stable matching which is not job-optimal, then there would be some candidate C matched with a job which does not propose to them in the propose-and-reject algorithm. We can demonstrate this with the following preferences lists. For any $n \geq 2$, let $J_1, \ldots, J_n$ be the jobs and $C_1, \ldots, C_n$ be the candidates.

The candidates' preferences lists are as follows.

Candidates	Jobs								
C_1	J_1	$\succ$	J_2	$\succ$	J_3	$\succ$	$\ldots$	$\succ$	J_n
C_2	J_2	$\succ$	J_1	$\succ$	J_2	$\succ$	$\ldots$	$\succ$	J_n
C_3	J_3	$\succ$	J_1	$\succ$	J_2	$\succ$	$\ldots$	$\succ$	J_n
$\vdots$					$\vdots$				
C_n	J_n	$\succ$	J_1	$\succ$	J_2	$\succ$	$\ldots$	$\succ$	J_{n-1}

And the jobs' preferences lists are as follows.

Jobs	Candidates								
J_1	C_n	$\succ$	C_{n-1}	$\succ$	C_{n-2}	$\succ$	$\ldots$	$\succ$	C_1
J_2	C_{n-1}	$\succ$	C_n	$\succ$	C_{n-1}	$\succ$	$\ldots$	$\succ$	C_1
J_3	C_{n-2}	$\succ$	C_n	$\succ$	C_{n-1}	$\succ$	$\ldots$	$\succ$	C_1
$\vdots$					$\vdots$				
J_n	C_1	$\succ$	C_n	$\succ$	C_{n-1}	$\succ$	$\ldots$	$\succ$	C_2

With these preferences lists, with the propose-and-reject algorithm, the jobs all make offers to distinct candidates on the first day; every candidate gets an offer and thus the algorithm terminates with the matching

$$\{(J_1, C_n), (J_2, C_{n-1}), \ldots, (J_n, C_1)\}.$$

However, with the flipped propose-and-reject algorithm, the candidates all make offers to distinct jobs on the first day; every job gets an offer and thus the algorithm terminates with the matching

$$\{(J_1, C_1), (J_2, C_2), \ldots, (J_n, C_n)\}.$$

This shows that for every stable matching with $n \geq 2$ jobs and candidates, there is a stable matching where jobs are matched with candidates to whom they never made an offer in the propose-and-reject algorithm.

(c) We disprove the statement. The flipped propose-and-reject algorithm will yield a candidate-optimal matching in which every candidate is matched with their optimal job. However, a single job cannot be matched with two candidates, so any two candidates must have different optimal jobs.

2 Best-Case and Worst-Case Scenarios

(a) We disprove the statement by contradiction. Suppose for contradiction that every job gets their matched with their least preferred candidate. For this to occur, every job must have a different least preferred candidate; if two jobs had the same least preferred candidate, only one could be matched with that candidate. The other would get a candidate other than their least preferred, which contradicts our assumption.

Let J_1 and J_2 be two jobs with least preferred candidates C_1 and C_2. Then the algorithm assigns (J_1, C_1) and (J_2, C_2). But then we can simply swap the candidates and keep the matching stable: (J_1, C_2) and (J_2, C_2). This cannot form any rogue couples since J_1 prefers C_2 to C_1 and J_2 prefers C_1 to C_2. This contradicts the fact that the propose-and-reject algorithm is job-optimal. So such a matching cannot have happened.

Numerous other approaches work as well, most using some form of contradiction. Consider the following, slightly less formal proof which disproves the claim by finding a contradiction with the fact that there is always at least one candidate who only receives one offer.

> Suppose for contradiction that in the propose-and-reject algorithm, each job is matched with their least preferred candidate.
>
> For this to occur, it must be that every candidate received an offer from every single job. To prove this, suppose for contradiction that some candidate C never received an offer from job J. Then some candidate C' which job J prefers to C must have received an offer from J and never rejected it. But then J must have been matched with C', which they prefer to C. This contradicts our assumption that every job is matched with their least preferred candidate.
>
> So every candidate receives an offer from every single job. However, we know from Discussion 2A, Question 3**(c)** that in any execution of the propose-and-reject algorithm, there is at least one candidate who only receives a single offer. But we have shown that every candidate receives an offer from all $n \geq 2$ jobs. This is a contradiction, so our assumption that every job was matched with their least preferred candidate must be incorrect. So no such set of preferences can exist.

(b) We prove the statement by example. Let $J_1, \dots J_n$ be the jobs and let $C_1, \dots, C_n$ be the candidates.

Consider the following preference lists for the jobs.

Jobs	Candidates								
J_1	C_1	$\succ$	C_2	$\succ$	C_3	$\succ$	$\dots$	$\succ$	C_n
J_2	C_2	$\succ$	C_1	$\succ$	C_2	$\succ$	$\dots$	$\succ$	C_n
J_3	C_3	$\succ$	C_1	$\succ$	C_2	$\succ$	$\dots$	$\succ$	C_n
$\vdots$					$\vdots$				
J_n	C_n	$\succ$	C_1	$\succ$	C_2	$\succ$	$\dots$	$\succ$	C_{n-1}

And consider the following preference lists for the candidates.

Candidates	Jobs								
C_1	J_n	$\succ$	$\dots$	$\succ$	J_3	$\succ$	J_2	$\succ$	J_1
C_2	J_n	$\succ$	$\dots$	$\succ$	J_3	$\succ$	J_1	$\succ$	J_2
C_3	J_n	$\succ$	$\dots$	$\succ$	J_2	$\succ$	J_1	$\succ$	J_3
$\vdots$					$\vdots$				
C_n	J_{n-1}	$\succ$	$\dots$	$\succ$	J_3	$\succ$	J_1	$\succ$	J_n

Since every job makes an offer to a different candidate, every candidate receives an offer and the algorithm terminates on the first day with the following stable matching:

$$\{(J_1, C_1), (J_2, C_2), \dots, (J_n, C_n)\}.$$

This matches each candidate with their least preferred job.

(c) We prove the statement by contradiction. Suppose that job J is candidate C's most preferred job and candidate C is job J's most preferred candidate, but they are not matched in some stable matching S.

Therefore there must be some job $J' \neq J$ and some candidate $C' \neq C$ such that $(J, C'), (J', C) \in S$. However, J prefers C over C' since C is J's most preferred candidate. Likewise, C prefers J over J', since J is C's most preferred job. Therefore (J, C) is a rogue couple in S. This contradicts that fact that S is stable.

Our assumption that there was a stable matching S where J and C are not paired was incorrect. Thus we have shown that if J and C are each other's most preferred partners, they will be matched. in any stable matching.

(d) We disprove the statement by counterexample. Consider the following example with $n = 2$ jobs and candidates.

Jobs	Candidates
J_1	$C_1 \succ C_2$
J_2	$C_1 \succ C_2$

Candidates	Jobs
C_1	$J_1 \succ J_2$
C_2	$J_1 \succ J_2$

In this case, in the propose-and-reject algorithm, both jobs make offers to candidate C_1. C_1 will reject J_2, and so the next day J_1 will make an offer to C_1 and J_2 will make an offer to C_2, resulting in the following stable matching:

$$\{(J_1, C_1), (J_2, C_2)\}.$$

Here, J_2 and C_2, who are each other's least preferred partners, are matched.

Based off of this example, we can see that this happens if some job J and candidate C are at the bottom of every candidate's and every job's preference list, respectively. This allows us to construct a general counterexample.

We construct our jobs' preferences lists so that they all share the same least preferred candidate.

Jobs	Candidates								
J_1	C_1	$\succ$	C_2	$\succ$	C_3	$\succ$	$\ldots$	$\succ$	C_n
J_2	C_2	$\succ$	C_1	$\succ$	C_3	$\succ$	$\ldots$	$\succ$	C_n
J_3	C_3	$\succ$	C_1	$\succ$	C_2	$\succ$	$\ldots$	$\succ$	C_n
$\vdots$					$\vdots$				
J_n	C_1	$\succ$	C_2	$\succ$	C_3	$\succ$	$\ldots$	$\succ$	C_n

And we construct our candidates' preferences lists so that they all share the same least preferred job.

Candidates	Jobs								
C_1	J_1	$\succ$	J_2	$\succ$	J_3	$\succ$	$\ldots$	$\succ$	J_n
C_2	J_2	$\succ$	J_1	$\succ$	J_3	$\succ$	$\ldots$	$\succ$	J_n
C_3	J_3	$\succ$	J_1	$\succ$	J_2	$\succ$	$\ldots$	$\succ$	J_n
$\vdots$					$\vdots$				
C_n	J_1	$\succ$	J_2	$\succ$	J_3	$\succ$	$\ldots$	$\succ$	J_n

On the first day, job J_i makes an offer to candidate C_i for each $i \in \{1, \ldots, n-1\}$. Moreover, J_i is C_i's most preferred candidate, so by the improvement lemma, they will be matched.

However, J_n will make an offer to C_1 on the first day and get rejected, to C_2 on the second day and get rejected, and so on. Meanwhile, C_n will not receive any offers until day n, where J_n makes an offer to C_n and the algorithm terminates with the following stable matching.

$$\{(J_1, C_1), (J_2, C_2), \ldots, (J_n, C_n)\}.$$

Here, J_n and C_n are paired even though they are each other's least preferred partners. So we have disproven the statement.

3 Counting Cartesian Products

(a) As shown in lecture, $\mathbb{N} \times \mathbb{N}$ is countable by creating a zigzag map which enumerates through the pairs of natural numbers:

$$(0,0), (1,0), (0,1), (2,0), (1,1), \ldots$$

. Since A is countable, there must exist some bijection $f : A \to S_A$ between A and a subset S_1 of $\mathbb{N}$. Likewise, since B is countable, there must exist some bijection $g : B \to S_B$ between B and a subset S_B of $\mathbb{N}$.

We consider the function $h : A \times B \to S_A \times S_B$, defined as follows:

$$h(a, b) \coloneqq (f(a), g(b)).$$

This function is well defined, since for any $a \in A$, $f(a) \in S_A$ and for any $b \in B$, $g(b) \in S_B$.

We claim h is a bijection. We must show that it is both an injection and a surjection.

h is an injection. Suppose that $h(a_1, b_1) = h(a_2, b_2)$. That is, $(f(a_1), g(a_1)) = (f(a_2), g(a_2))$. Therefore $f(a_1) = f(a_2)$ and $g(a_1) = g(a_2)$. But f is a bijection and thus an injection, so $a_1 = a_2$. Similarly, since g is a bijection and thus an injection, $b_1 = b_2$. Thus we have that $(a_1, b_1) = (a_2, b_2)$, as desired.

h is a surjection. Consider any $(s_a, s_b) \in S_A \times S_B$. Since f is a bijection and thus a surjection, there exists some a such that $f(a) = s_a$. Similarly, since g is a bijection and thus a surjection, there exists some b such that $g(b) = s_b$. Therefore

$$h(a, b) = (f(a), g(b)) = (s_a, s_b),$$

so we have found an (a, b) such that $h(a, b) = (s_a, s_b)$, as desired.

We have demonsrated a bijection from $A \times B$ to $S_A \times S_B \subseteq \mathbb{N} \times \mathbb{N}$. Since $\mathbb{N} \times \mathbb{N}$ is countable, we are done.

(b) By induction on n.

Base cases.

$n = 1$. If A_1 is countable, then

$$\bigtimes_{i=1}^{1} A_i = A_1$$

is also countable.

$n = 2$. By part **(a)**, if A_1 and A_2 are both countable, then $A_1 \times A_2$ is countable.

Induction case.

Induction hypothesis. Suppose that for some $n \in \mathbb{N}$,

$$\bigtimes_{i=1}^{n} A_i = A_1 \times \ldots \times A_n$$

is countable.

Induction step. Consider

$$\bigtimes_{i=1}^{n+1} A_i = \left(\bigtimes_{i=1}^{n} A_i \right) \times A_{n+1}.$$

By the induction hypothesis,

$$A = \bigtimes_{i=1}^{n} A_i$$

is countable. By the $n = 2$ base case,

$$\bigtimes_{i=1}^{n+1} A_i = A \times A_{n+1}$$

is countable since both A and A_{n+1} are countable.

(c) We demonstrate an injection from $\{0, 1\}^\star$, the set of infinite bit strings to our set.

For each $i \in \mathbb{N}^+$, enumerate $B_i = \{B_{i0}, B_{i1}, \ldots\}$, where every set has at least two elements B_{i0} and B_{i1}. Consider the function

$$f : \{0, 1\}^\star \to \bigtimes_{i=1}^{\infty} B_i$$

defined as follows: for each infinite binary string $b_0 b_1 b_2 \ldots \in \{0, 1\}^\star$, define

$$f(b_0 b_1 b_2 \ldots) \coloneqq (B_{1b_0}, B_{1b_1}, B_{2b_2}, \ldots).$$

This is an injection. Consider two different infinite binary strings $b_0 b_1 b_2 \ldots$ and $c_0 c_1 c_2 \ldots$. Since they are different, there must be some $i \in \mathbb{N}$ such that $b_i \neq c_i$, e.g. $b_i = 1$ and $c_i = 0$ or $b_i = 0$ and $c_i = 1$. Then $B_{ib_i} \neq B_{ic_i}$, since $B_{i0} \neq B_{i1}$. Therefore

$$f(b_0 b_1 b_2 \ldots b_i \ldots) = (B_{1b_0}, B_{1b_1}, B_{2b_2}, \ldots, B_{ib_i}, \ldots) \neq (B_{1c_0}, B_{1c_1}, B_{2c_2}, \ldots, B_{ic_i}, \ldots).$$

So f is an injection. Therefore

$$|\mathbb{R}| = |\{0, 1\}^\star| \leq \left| \bigtimes_{i=1}^{\infty} B_i \right|.$$

That is, the set is uncountable, since its cardinality is at least the cardinality of the real numbers.

Alternatively, we can prove that the set is uncountable by diagonalization. Suppose for contradiction that $S = B_1 \times B_2 \times$ is countable and it can be enumerated. That is, we can write $S = \{s_1, s_2, s_3, s_4, \ldots\}$. Then we can write the enumeration as follows

	1	2	3	4	...
s_1	$(b_{11}$	b_{21}	b_{31}	b_{41}	$\ldots)$
s_2	$(b_{12}$	b_{22}	b_{32}	b_{42}	$\ldots)$
s_3	$(b_{13}$	b_{23}	b_{33}	b_{43}	$\ldots)$
s_4	$(b_{14}$	b_{24}	b_{34}	b_{44}	$\ldots)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

where b_{ij} represents the item from B_i which is included is the j^{th} enumerated element of S.

Consider the element

$$s = (\bar{b}_{11}, \bar{b}_{22}, \bar{b}_{33}, \bar{b}_{44} \ldots),$$

where $\bar{b}_{ii}$ is any element from B_i which is not b_{ii}. Such an element of each B_i exists since every B_i has at least two elements.

Then $s \in S$, since it is an infinite sequence of elements from B_1, B_2, and so on. However, by construction, s is not the same as any elements in the numerator $\{s_1, s_2, s_3, \ldots\}$, since it differs from s_k at the k^{th} position, where $\bar{b}_{kk} \neq b_{kk}$ by construction. So $s \notin S$. This is a contradiction, so our assumption that S could be enumerated was incorrect.

We have shown that S is uncountable.

4 Counting Shapes

(a) We prove that every collection of filled squares is countable. We will show an injection from any collection $\mathcal{C}$ to $\mathbb{Q} \times \mathbb{Q}$. Here, we know $\mathbb{Q} \times \mathbb{Q}$ to be countable since $\mathbb{Q}$ is countable, and by Question 3, finite Cartesian products of countable sets are countable.

For any filled square $F \in \mathcal{C}$, we can find a point $(p_F, q_F) \in F$ in F with rational coordinates. Let $2r_F \in \mathbb{R}$ be the side length of F and let $(x_F, y_F) \in F$ be the center of F. Then

$$F = [x_F - r_F, x_F + r_F] \times [y_F - r_F, y_F \times r_F].$$

However, any interval of the real contains a rational number (in fact, infinitely many), so there is some rational $p_F \in [x_F - r_F, x_F + r_F]$ and there is some rational $q_F \in [y_F - r_F, y_F + r_F]$. Therefore $(p_F, q_F) \in F$, but also $(p_F, q_F) \in \mathbb{Q} \times \mathbb{Q}$.

Once we have found such a rational point for each filled square in $\mathcal{C}$, we define the function $f : \mathcal{C} \to \mathbb{Q} \times \mathbb{Q}$ as $f(F) \coloneqq (p_F, q_F)$ for every $F \in \mathcal{C}$.

We claim f is an injection. Suppose that for $F_1, F_2 \in \mathcal{C}$, $f(F_1) = f(F_2)$. That is, $(p_1, q_1) = (p_2, q_2)$, where $(p_1, q_1) \in F_1$ and $(p_2, q_2) \in F_2$ are the rational points assigned to F_1 and F_2. Call this one point $(p, q) = (p_1, q_1) = (p_2, q_2)$.

That is, F_1 and F_2 both contain the rational point (p, q). Then it cannot be that $F_1 \neq F_2$, since F_1 and F_2 must be disjoint if they are two distinct elements of $\mathcal{C}$. It must be that $F_1 = F_2$. So f is an injection.

Therefore we have shown that for any $\mathcal{C}$,

$$|\mathcal{C}| \leq |\mathbb{Q} \times \mathbb{Q}| = |\mathbb{N}|.$$

That is, any collection $\mathcal{C}$ must be countable, since it is at most as large as a countable set.

(b) We prove that there exists a collection $\mathcal{C}$ of empty squares which is uncountable.

For each $r \in \mathbb{R}$, we have a square $V_r \in \mathcal{C}$ with corners at

$$(-r/2, -r/2), \quad (r/2, -r/2), \quad (-r/2, r/2), \quad (r/2, r/2).$$

That is, V_r is the square with side length r centered at the origin. Then no two squares in $\mathcal{C}$ intersect.

Consider the function $f : \mathbb{R} \to \mathcal{C}$ defined as $f(r) = V_r$ for each $r \in \mathbb{R}$. We claim this is an injection. Suppose that we have two different real numbers $r, s \in \mathbb{R}$. Then, by construction, $f(r) = V_r \neq V_s = f(s)$, since V_r has side length r and V_s has side length s.

Since f is an injection,

$$|\mathbb{R}| \leq |\mathcal{C}|,$$

so $\mathcal{C}$ is uncountable because it is at least as large as an uncountable set.

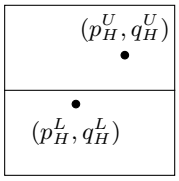
(c) We prove that every collection of halved empty squares is countable. As in **(a)**, we will show an injection from any collection $\mathcal{C}$ to a countable set.

To determine what to inject any given collection to, we need to find some "information" about any given halved empty square in the collection which distinguishes it from every other halved empty square.

Note that while a rational point exists inside the square, that is not enough to distinguish between two halved empty squares because halved empty squares can be inside one another. Rather, we may want to consider using more information than just one rational point: instead, we could use two rational points.

Many constructions will work, such as considering a rational point inside and a rational point outside a given halved empty square. Here, we will consider placing a rational point in each the top and bottom halves of the square.

For any halved empty square $H \in \mathcal{C}$, we can find a point (p_H^U, q_H^U) in the upper half of H with rational coordinates, and we can likewise find a point (p_H^L, q_H^L) in the lower half of H with rational coordinates. The process is much the same as in **(a)**.



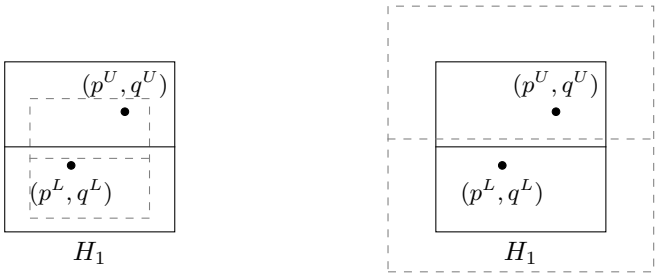
Once we have a pair of rational points for each halved empty square in $\mathcal{C}$, we define the function $f : \mathcal{C} \to (\mathbb{Q}\times\mathbb{Q})\times(\mathbb{Q}\times\mathbb{Q})$ as $f(H) \coloneqq ((p_H^U, q_H^U), (p_H^L, q_H^L))$. By Question 2, $(Q \times Q) \times (\mathbb{Q} \times \mathbb{Q}) = \mathbb{Q}^4$ is countable since it is a finite Cartesian product of countable sets.

We claim that f is an injection. Suppose that for $H_1, H_2 \in \mathcal{C}$, $f(H_1) = f(H_2)$. That is,

$$((p_1^U, q_1^U), (p_1^L, q_1^L)) = ((p_1^U, q_1^U), (p_1^L, q_1^L)).$$

Call this pair of points $((p^U, q^U), (p^L, q^L))$.

If H_2 has a point in its upper half which is in the upper half of H_1, and H_2 has a point in its lower half which is in the lower half of H_1, it must necessarily intersect with H_1.



Since $\mathcal{C}$ consists of disjoint shapes, it must be that $H_1 = H_2$. So f is an injection.

Therefore for any collection $\mathcal{C}$,

$$|\mathcal{C}| \leq |\mathbb{Q}^4| = |\mathbb{N}|,$$

so any $\mathcal{C}$ is countable since it has cardinality at most that of a countable set.

5 Edge Complement

(a) We prove the statement via a direct proof. Consider any edge $\{i, j\} \in E$ in G. If G has an Eulerian tour, then both the vertices i and j had an even number of incident edges. In particular, besides $\{i, j\}$, the vertex i had an odd number of other incident edges $\{i, k\}$ for $k \neq j$ and the vertex j had an odd number of other incident edges $\{j, \ell\}$ for $\ell \neq i$.

In G', the edge $\{i, j\}$ becomes a vertex. We show that it must have an even number of edges. Note that $\{i, j\}$ gets an edge for each of the odd number of vertices $k \neq j$ which i was adjacent to. Likewise, $\{i, j\}$ gets an edge for each of the

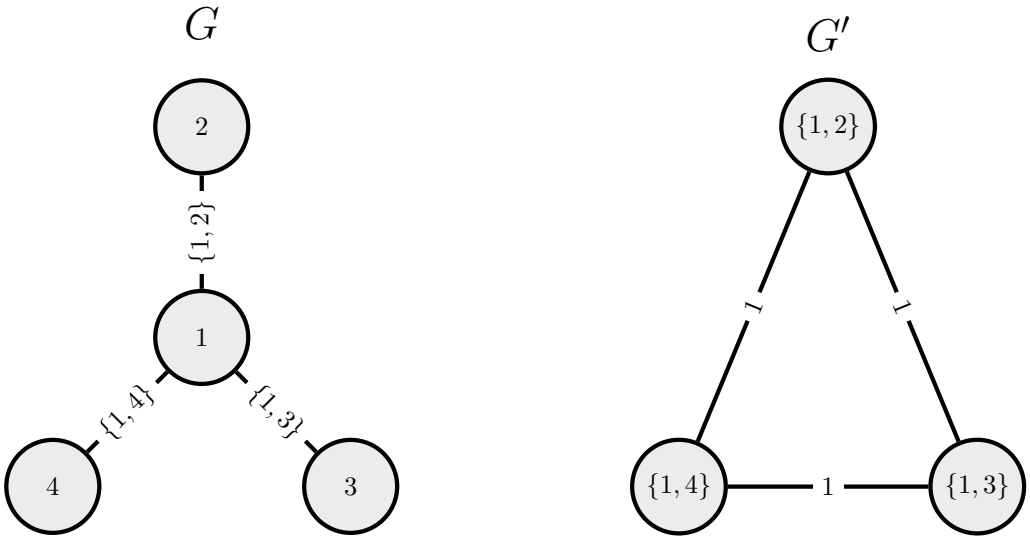
odd number of vertices $\ell \neq i$ which j was adjacent to. The vertex $\{i, j\}$ gets no additional edges, since it can only form an edge with another vertex if that vertex represents an edge containing one of i and j.

Therefore the vertex $\{i, j\}$ has an even number of edges in G', since the sum of two odd numbers is even. However, this vertex was arbitrary. So every vertex in G' has an even number of edges. Therefore G' has an Eulerian tour.

(b) We disprove the statement. If there is an Eulerian tour in G', then every vertex $\{i, j\}$ will have an even number of neighbors. This means that in G, there are either an odd number of edges besides $\{i, j\}$ incident to each i and j or there are an even number of edges besides $\{i, j\}$ incident to each i and j. If, for example, in G there were an even number of edge n_i incident to i but an odd number of edges n_j incident to j, then in G', the vertex $\{i, j\}$ would have $n_i + n_j$ neighbors, which would be an odd number. So such a situation cannot happen.

If i and j each have an odd number of edges besides $\{i, j\}$ incident to them, then they each have even degree. However, if i and j each have an even number of edges besides $\{i, j\}$ incident to them, then they each have odd degree. In such a situation, G will not have an Eulerian tour.

We can use the above idea to construct our counterexample. We need to create edges $\{i, j\}$ such that i and j both have odd degree. Consider the following graph G and its edge complement graph G'.



The graph G' has an Eulerian tour since all its vertices have even degree. However, the graph G does not have an Eulerian tour since the vertex 1 has odd degree.

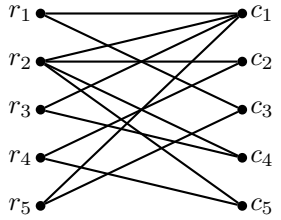
6 Filling the grid

(a) We use a direct proof. Suppose that there are an odd number of blanks in a row or column. That is, let the number of blanks in that row or column be $2k + 1$ for some $k \in \mathbb{N}$. Then the number of $+1$s in that column must be between 0 and $2k + 1$. In particular, for $n_+ \in \{0, 1, \ldots, 2k, 2k + 1\}$ the number of $+1$s in that row or column, the number of -1s in that row or column will be $n_- = (2k + 1) - n_+$. Therefore the sum for that row or column will be

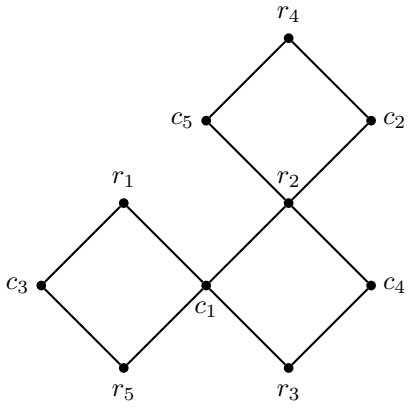
$$\sum_{i=1}^{n_+} 1 + \sum_{i=1}^{n_-} (-1) = n_+ - n_- = n_+ - (2k + 1 - n_+) = 2(n_+ - k) - 1.$$

For this sum to be 0, we require that $n_+ - k = 1/2$. Since $k, n_+ \in \mathbb{N}$ are both natural numbers, this is impossible. So if there are an odd number of blanks in a row or column, then there is no way to fill the grid.

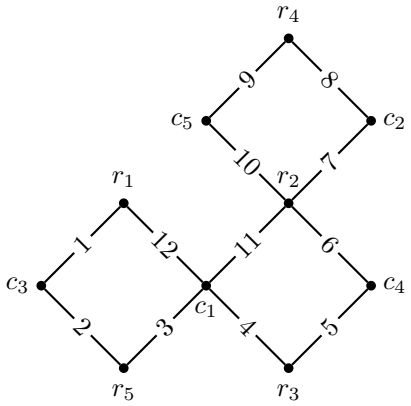
(b) Simply drawing the vertice and edges yields the following graph.



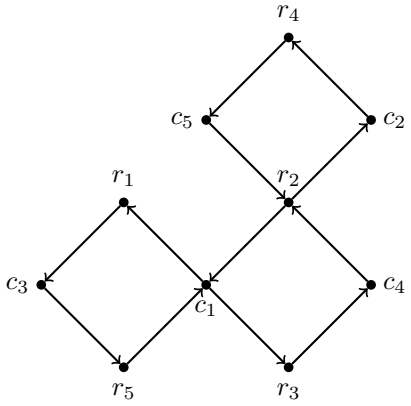
However, drawing out the graph in a more reasoned manner results in the following drawing, which is easier to work with.



(c) There are any. Here is one possible Eulerian tour.



(d) We use the Eulerian tour from **(c)**.



(e) For $v \in \{r_1, c_3, r_5, r_3, c_4, c_2, r_4, c_5\}$ we have that

$$\deg^-(v) - \deg^+(v) = 1 - 1 = 0.$$

For $v \in \{c_1, r_2\}$, we have that

$$\deg^-(v) - \deg^+(v) = 2 - 2 = 0.$$

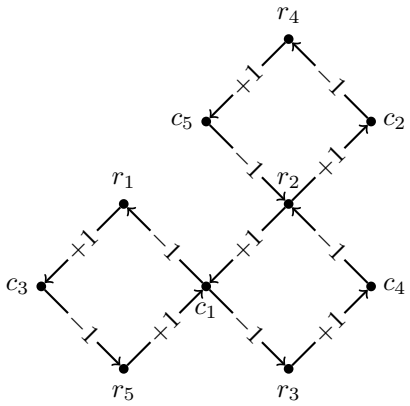
The difference is always zero; that is, the in-degree and out-degree are the same.

(f) If an edge in G' has direction $r_i \to c_j$, we fill in the corresponding block with $+1$. If the edge instead has direction $c_j \to r_i$, we fill in -1.

To show that this works, we must show that each row and column sum up to zero. For a Eulerian tour, we know that it must enter and exit each vertex the same number of times. For each row vertex r_i we know that our tour enters it $\deg^-(r_i)$ times and exits it $\deg^+(r_i)$ times. Thus we must have $\deg^-(r_i) = \deg +(r_i)$, which validates our observation from **(e)**. The same is true for each column vertex c_j.

Then, for each row, because the edges exiting it correspond to $+1$ and edges entering it correspond to -1, we know the total sum is $\deg^+(r_i) - \deg^-(r_i) = 0$. For each column, the opposite is true: the total sum is $\deg^-(c_j) - \deg^+(c_j) = 0$. Thus, the total sum is always zero for each row and column.

When we apply this to our graph G', we get



This yields the following solution to Aliyah's 5×5 grid.

-1		$+1$		
$+1$	$+1$		-1	-1
-1			$+1$	
	-1			$+1$
$+1$		-1		

7 Planarity and Graph Complements

(a) If G has v vertices, then there are a total of $\frac{v(v-1)}{2}$ edges that could possibly exist in the graph. Since e of them appear in G, we know that the remaining $\frac{v(v-1)}{2} - e$ must appear in $\overline{G}$.

(b) Since G is planar, we know that $e \leq 3v - 6$. Plugging this in to the answer from the previous part, we have that $\overline{G}$ has at least $\frac{v(v-1)}{2} - (3v - 6)$ edges. Since v is at least 13, we have that $\frac{v(v-1)}{2} \geq \frac{v \cdot 12}{2} = 6v$, so $\overline{G}$ has at least $6v - 3v + 6 = 3v + 6$ edges. Since this is strictly more than the $3v - 6$ edges allowed in a planar graph, we have that $\overline{G}$ must not be planar.

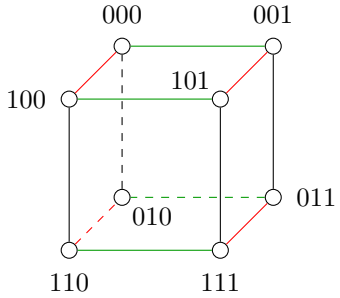
(c) The converse is not necessarily true. As a counterexample, suppose that G has exactly 13 vertices, of which five are all connected to each other and the remaining ten have no edges incident to them. This means that G is non-planar, since it contains a copy of K_5. However, $\overline{G}$ also contains a copy of K_5 (take any 5 of the 8 vertices that were isolated in G), so $\overline{G}$ is also non-planar. Thus, it is possible for both G and $\overline{G}$ to be non-planar.

8 Hypercubes

(a) The three hypercubes are a line, a square, and a cube, respectively. See also note 5 for pictures.

(b) Consider each edge that changes the ith bit for some $i \leq n$. Every vertex touches exactly one of these edges, because there is exactly one way to change the ith bit in any bitstring. Coloring each of these edges color i ensures that each vertex will then be adjacent to n differently colored edges, since there are n different bits to change, and no two edges representing bit changes on different bits have the same color.

An example for the three dimensional case is shown below (red is the first bit, blue is the second bit, and green is the third bit):



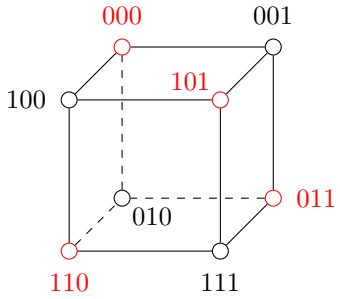
Alternate solution (using induction):

In the base case of $n = 1$, the hypercube of only one line can be edge colored with 1 color. Next, suppose that the n dimensional hypercube can be colored with n colors. Recall that the $n + 1$ dimensional hypercube is composed of two n dimensional hypercubes; each of these hypercubes can be colored with n colors by the inductive hypothesis.

We can connect the two n dimensional hypercubes with edges colored with a different color; this will be our $(n + 1)$th color. Since these new edges will always be between distinct pairs of vertices, one from each subcube, none of these new edges will share a vertex, giving a valid coloring of the $n + 1$ dimensional hypercube with $n + 1$ colors.

(c) Consider the vertices with an even number of 0 bits and the vertices with an odd number of 0 bits. Each vertex with an even number of 0 bits is adjacent only to vertices with an odd number of 0 bits, since each edge represents a single bit change (either a 0 bit is added by flipping a 1 bit, or a 0 bit is removed by flipping a 0 bit). Let L be the set of the vertices with an even number of 0 bits and let R be the vertices with an odd number of 0 bits, then no two adjacent vertices will belong to the same set.

An example for the three dimensional case is shown below (L are blue vertices, and R are red vertices):



Alternate solution (using induction and coloring):

It may be simpler to that a graph being 2-colorable is the same as being bipartite. Now, the argument is easier to state. First the base case is a hypercube with two vertices which is clearly two-colorable. Then notice, switching the colors in a two-coloring is still valid as if endpoints are differently colored, switching leaves them differently colored. Now, recursively one two colors the two subcubes the same, and then switches the colors in one subcube. The internal to subcube edges are fine by induction. The edges across are fine as the corresponding vertices are differently colored due to the switching.