CS 70 Discrete Mathematics and Probability Theory
Summer 2017 Hongling Lu, Vrettos Moulos, and Allen Tang **HW 2**

Sundry

Before you start your homework, write down your team. Who else did you work with on this homework? List names and email addresses. (In case of homework party, you can also just describe the group.) How did you work on this homework? Working in groups of 3-5 will earn credit for your "Sundry" grade.

Please copy the following statement and sign next to it:

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up.

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up. (Signature here)

1 Optimal Partners

In the notes, we proved that the SMA always outputs the male-optimal pairing. However, we never explicitly showed why it is guaranteed that putting every man with his best choice results in a pairing at all. Prove by contradiction that no two men can have the same optimal partner. (Note: your proof should not rely on the fact that the SMA outputs the male-optimal pairing.)

Solution:

For the sake of contradiction, assume that we have some instance of the Stable Marriage problem where both man M and man M' have woman W as their optimal partner. We further assume without loss of generality that W prefers M to M' (if this is not the case, we can just switch the names to make it so). Because W is M''s optimal partner, we know by definition that there must exist some stable pairing in which M' is paired with W. Call M's partner in that pairing W^*. Since W is M's optimal partner, we know by definition that M must prefer W to any woman he is ever paired with in any stable pairing–including W^*. As well, we previously said that W prefers M to M'. Thus, M and W would form a rouge couple in this pairing, which contradicts our statement that the pairing we were dealing with was stable. So our initial assumption must be false: there must never exist two men who have the same optimal partner.

2 Relaxed Timing

Suppose that when running the SMA, we relax the rules for the men, so that each unpaired man proposes to the next woman on his list at a time of his choice (some men might procrastinate for several days, while others might propose and get rejected several times in a single day). Can the order of the proposals change the resulting pairing? Give an example of such a change or prove that the pairing that results is the same.

Solution:

Assume, that when a proposal is made and an answer is received, we write down it on a list L and enumerate them. Now the proof is similar to the proof in class that the algorithms finds male optimal pairing.

We should prove that the pairing P' that results is the same as pairing P found by the regular algorithm. Assume the opposite, so either there is a woman that rejects a man who was her pair in P or there is a woman who do not receive a proposal from a man who was her pair in P. By Well Ordering Principle, there is the first entry in the list L for one of that happens, let W denote the woman for whom it happens and let M denote her pair in P.

First, if W rejected M then it is because she get proposal from other man M' who is higher in her preference list. But M' proposed her only because he was rejected by other woman W' who is higher in his preference list and did not rejects him in P. In other words, even before W rejected M, W' rejected her previous pair M'. Contradiction.

Second, if W did not get a proposal from M then it is because M was not rejected by some other woman W' who is higher in his preference list and who rejects him in P. Contradiction.

Therefore, the resulting pairing P' is the same as P.

3 Connectivity

Prove the following claims regarding connectivity:

(a) If G is a graph with n vertices such that for any two non-adjacent vertices u, v, it holds that $\deg u + \deg v \geq n-1$, then G is connected.

(b) Give an example to show that if the condition $\deg u + \deg v \geq n-1$ is replaced with $\deg u + \deg v \geq n-2$, then G is not necessarily connected.

(c) For a graph G with n vertices, if the minimum degree of each vertex is at least $n/2$, then G is connected.

(d) If there are exactly two vertices with odd degrees in a graph, then they must be connected to each other (meaning, there is a path connecting these two vertices).

Solution:

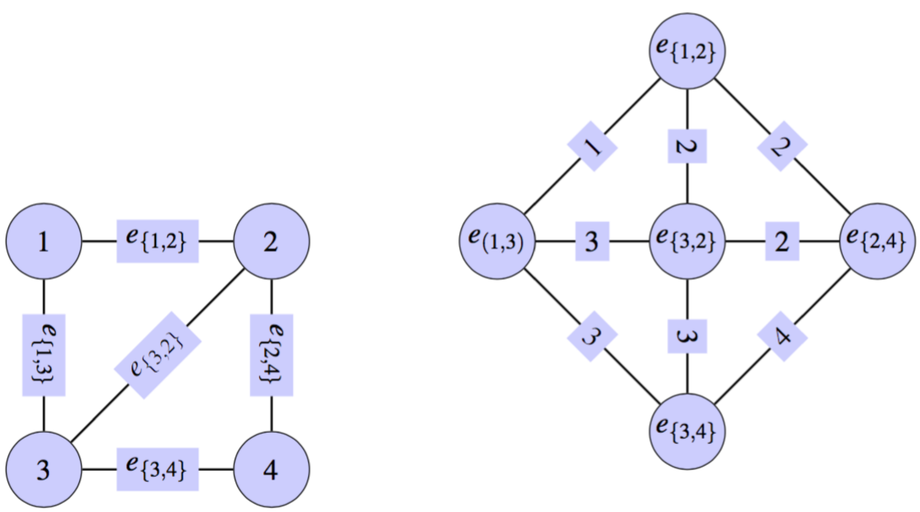
(a) Consider non-adjacent u and v. Then, there must be a vertex w such that u and v are both adjacent to w. To see why, suppose this is not the case. Then, the set of neighbors of u and v has $n-1$ elements, but there are only $n-2$ other vertices. (This is the Pigeonhole Principle.) We have proven that for any non-adjacent u and v, there is a path $u \to w \to v$, and thus G is connected.

(b) Consider two disconnected copies of K_n. There are $2n$ vertices total. For non-adjacent u, v, it holds that $\deg u + \deg v = (n-1) + (n-1) = 2n-2$, but the entire graph is not connected.

(c) Suppose that G is not connected. There must be at least two connected components, say G_1 and G_2. One of them will have at most $n/2$ vertices, and the maximum degree in this subgraph will be at most $n/2 - 1$. A contradiction.

(d) Suppose that they are not connected to each other. Then they must belong to two different connected components, say G_1 and G_2. Each of them will only have one vertex with odd degree. This leads to a contradiction since the sum of all degrees should be an even number.

4 Edge Complement



The **edge complement** graph of a graph $G = (V, E)$ is a graph $G' = (V', E')$, such that $V' = E$, and $(i, j) \in E'$ if and only if i and j had a common vertex in G. In the above picture, the graph on the right is the edge complement of the graph on the left: for every edge $e_{\{i,j\}}$ in the graph on the left there is a vertex in the graph on the right. If two edges $e_{\{i,j\}}$ and $e_{\{j,k\}}$ share a vertex j on the left, then the corresponding vertices on the right have an edge j connecting them.

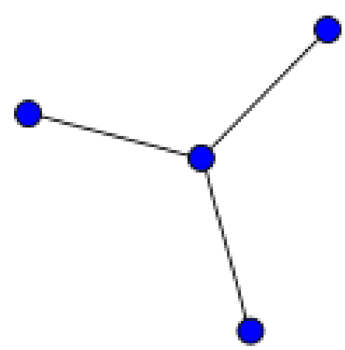
(a) Prove or disprove: if a graph G has an Eulerian tour, then its **edge complement** graph has an Eulerian tour.

(b) Prove or disprove: if a graph's **edge complement** graph G' has an Eulerian tour, then graph G has an Eulerian tour.

Solution:

(a) **True.** Using the same notation as above, we let an edge in G be $e_{\{i,j\}}$, with endpoints i and j. Then in G', $e_{\{i,j\}}$ is a vertex with edges between itself and only those vertices whose edge-representations in G shared the same vertex as itself. In other words, $e_{\{i,j\}}$ in G' will be neighbors with vertices of the form $e_{\{a,i\}}$ for some $a \neq i$ and $e_{\{j,b\}}$ for some $b \neq j$. If G had an Eulerian tour, then both i and j were incident to an even number of edges; this means that besides $e_{\{i,j\}}$, there were an odd number of other edges in G which were also incident to i, and likewise an odd number of other edges also incident to j. Thus in G', $e_{\{i,j\}}$ has an $odd + odd = even$ number of neighbors and thus is incident on an even number of edges. This is true for all vertices in G'; therefore, there is an Eulerian tour in G'.

(b) **False.** We will again use the same notation as above. If there is an Eulerian tour in G', any vertex $e_{\{i,j\}}$ will have an even number of neighbors. This means that in G, there are either an odd number of other edges besides $e_{\{i,j\}}$ incident to both i and j, or there are an even number of other edges incident to both i and j; i.e. in order for $e_{\{i,j\}}$ to in total have an even number of adjacent edges in G, the size of its two groups of neighbors must be either both an even number or both an odd number. If both groups of neighbors are odd, then both i and j have even degrees, since we add $e_{\{i,j\}}$ to the each group to make up the set of incident edges to i and j. However, if the number of neighbors in both groups are even, then both i and j will have odd degrees, since again we must add $e_{\{i,j\}}$ to both groups. If this is the case, then G will not have an Eulerian Tour. Also, a counterexample is a star with 3 vertices.



5 Always, Sometimes, or Never

In each part below, you are given some information about a graph G. Using only the information in the current part, say whether G will always be planar, always be non-planar, or could be either. If you think it is always planar or always non-planar, prove it. If you think it could be either, give a planar example and a non-planar example.

(a) G can be vertex-colored with 4 colors.

(b) G requires 7 colors to be vertex-colored.

(c) $e \leq 3v - 6$, where e is the number of edges of G and v is the number of vertices of G.

(d) G is connected, and each vertex in G has degree at most 2.

(e) Each vertex in G has degree at most 2.

Solution:

(a) Either planar or non-planar. By the 4-color theorem, any planar graph can provide the planar example. The easiest non-planar example is $K_{3,3}$, which can be 2-colored because it is bipartite. (Certainly, any graph which can be colored using only 2 colors can also be colored using 4 colors.)

(b) Always non-planar. The 4-color theorem tells us that if a graph is planar, it can be colored using only 4 colors. The contrapositive of this is that if a graph requires more than 4 colors to vertex-color, it must be non-planar. (Using the 5- or 6-color theorem would also work.)

(c) Either planar or non-planar. From the notes, we know that every planar graph follows this formula, so any planar graph is a valid planar example. The easiest non-planar example is again $K_{3,3}$, which has $e = 9$ and $v = 6$, meaning our formula becomes $9 \leq 3(6) - 6 = 12$, which is certainly true.

(d) Always planar. There are two cases to deal with here: either G is a tree, or G is not a tree and so contains at least one cycle. In the former case, we're immediately done, since all trees are planar. In the latter case, consider any cycle in G. We know that every vertex in that cycle is adjacent to the vertex to its left in the cycle and to the vertex to its right in the cycle. But we also know that no vertex can be connected to more than two other vertices, so the cycle isn't connected to anything else. But G is a connected graph, so we must have that G is just a single large cycle. And we can certainly draw a simple cycle on a plane without crossing any edges, so even in this case G is still planar.

(e) Always planar. Each of G's connected components is connected and has no vertex of degree more than 2, so by the previous part, each of them must be planar. Thus, each of G's connected components must be planar, so G itself must be planar.

6 Bipartite Graphs

An undirected graph is bipartite if its vertices can be partitioned into two disjoint sets L, R such that each edge connects a vertex in L to a vertex in R (so there does not exist an edge that connects two vertices in L or two vertices in R).

(a) Suppose that a graph G is bipartite, with L and R being a bipartite partition of the vertices. Prove that $\sum_{v \in L} \deg(v) = \sum_{v \in R} \deg(v)$.

(b) Suppose that a graph G is bipartite, with L and R being a bipartite partition of the vertices. Let s and t denote the average degree of vertices in L and R respectively. Prove that $s/t = |R|/|L|$.

(c) A double of a graph G consists of two copies of G with edges joining the corresponding "mirror" points. Now suppose that G_1 is a bipartite graph, G_2 is a double of G_1, G_3 is a double of G_2, and so on. Show that $\forall n \geq 1$, G_n is bipartite.

(d) Prove that a graph is bipartite if and only if it can be 2-colored.

Solution:

(a) Since G is bipartite, each edge connects one vertex in L with a vertex in R. Since each edge contributes equally to $\sum_{v \in L} \deg(v)$ and $\sum_{v \in R} \deg(v)$, we see that these two values must be equal.

(b) By part (a), we know that $\sum_{v \in L} \deg(v) = \sum_{v \in R} \deg(v)$. Thus $|L| \cdot s = |R| \cdot t$. A little algebra gives us the desired result.

(c) We use induction. Let $P(n)$ be the proposition that G_n is bipartite. The base case is when $n = 1$. We see that $P(1)$ must be true by definition (and construction) of G_1. Now suppose that for $k \geq 1$, $P(k)$ holds. We see that the graph G_{k+1} consists of two subgraphs with the same structure as G_n, except the edges joining the corresponding edges of the two subgraphs. Remove the extra edges. Since $P(k)$ is true, we can label each vertex of one subgraph alternating between L and R such that the two are disjoint. If we color the corresponding vertices of the other subgraph with the opposite set, then the sets L and R will still be disjoint. Adding back the extra edges, we see that each edge connects a vertex in L to a vertex in R. Thus G_{k+1} is bipartite.

(d) Given a bipartite graph, color all of the vertices in L one color, and all of the vertices in R the other color. Conversely, given a 2-colored graph (call the colors red and blue), there are no edges between red vertices and red vertices, and there are no edges between blue vertices and blue vertices. Hence, take L to be the set of red vertices and R to be the set of blue vertices. We see that the graph is bipartite.

7 Countability Practice

(a) Prove or disprove: The set of increasing functions $f : \mathbb{N} \to \mathbb{N}$ (i.e., if $x \geq y$, then $f(x) \geq f(y)$) is countable.

(b) Prove or disprove: The set of decreasing functions $f : \mathbb{N} \to \mathbb{N}$ (i.e., if $x \geq y$, then $f(x) \leq f(y)$) is countable.

(c) Is a set of disks in $\mathbb{R}^2$ such that no two disks overlap necessarily countable or possibly uncountable? [A disk is a region in the plane of the form $\{(x,y) \in \mathbb{R}^2 : (x-x_0)^2 + (y-y_0)^2 \leq r^2\}$, for some $x_0, y_0, r \in \mathbb{R}$, $r > 0$.]

(d) Is a set of circles in $\mathbb{R}^2$ such that no two circles overlap necessarily countable or possibly uncountable? [*Hint*: A circle is a subset of the plane of the form $\{(x,y) \in \mathbb{R}^2 : (x-x_0)^2 + (y-y_0)^2 = r^2\}$ for some $x_0, y_0, r \in \mathbb{R}$, $r > 0$. The difference between a circle and a disk is that a disk contains all of the points in its interior, whereas a circle does not.]

Solution:

(a) Suppose that there is a bijection between $\mathbb{N}$ and the set of all increasing functions $\mathbb{N} \to \mathbb{N}$:

$$\begin{aligned}
0 &\mapsto (f_0(0), f_0(1), f_0(2), \ldots) \\
1 &\mapsto (f_1(0), f_1(1), f_1(2), \ldots) \\
2 &\mapsto (f_2(0), f_2(1), f_2(2), \ldots) \\
&\vdots
\end{aligned}$$

We will use a diagonalization argument to prove that there is a function f which is not in the above list. Define

$$f(n) = 1 + \sum_{i=1}^{n} f_i(n).$$

First, we will show that f is increasing. Indeed, if $m \le n$, then

$$f(m) = 1 + \sum_{i=1}^{m} f_i(m) \le 1 + \sum_{i=1}^{n} f_i(m) \le 1 + \sum_{i=1}^{n} f_i(n) = f(n).$$

The first inequality is because each function is non-negative; the second inequality is because the f_i are increasing.

To show that f is not in the list, note that

$$f(n) = 1 + \sum_{i=1}^{n} f_i(n) \ge 1 + f_n(n) > f_n(n).$$

Since $f(n) > f_n(n)$ for each $n \in \mathbb{N}$, f cannot be any of the functions in the list. Therefore, the set of increasing functions $f : \mathbb{N} \to \mathbb{N}$ is uncountable.

(b) Given any function that begins with $f(0) = n$, consider the number of indices in which the function decreases in output: the set of i such that $f(i) < f(i-1)$. There are only at most n such indices because eventually, the function will hit $f(n) = 0$ for which every subsequent input will output 1. We can set a bijection for any function with $f(0) = n$ to a "word" of indices at which the function decreases. Therefore, the set of decreasing functions $\mathbb{N} \to \mathbb{N}$ has the same cardinality as the set of finite bit strings from a countably infinite alphabet, which is countable. Therefore, the set of all decreasing functions is countable.

(c) Countable. Each disk must contain at least one rational point (an (x,y)-coordinate where $x, y \in \mathbb{Q}$) in its interior, and due to the fact that no two disks overlap, the cardinality of the set of disks can be no larger than the cardinality of $\mathbb{Q} \times \mathbb{Q}$, which we know to be countable.

(d) Possibly uncountable. Consider the circles $C_r = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 = r\}$ for each $r \in \mathbb{R}$. For $r_1 \neq r_2$, C_{r_1} and C_{r_2} do not overlap, and there are uncountably many of these circles (one for each real number).

8 Impossible Programs

Show that none of the following programs can exist.

(a) Consider a program P that takes in any program F, input x and output y and returns true if $F(x)$ outputs y and returns false otherwise.

(b) Consider a program P that takes in any program F and returns true if $F(F)$ halts and returns false if it doesn't halt.

(c) Consider a program P that takes in any programs F and G and returns true if F and G halt on all the same inputs and returns false otherwise.

Solution:

(a) If P exists, we can solve the halting problem. We show this by constructing HALT(F,x) where F is a program and x is the input. We will use P as a subroutine to derive a contradiction.

```
def HALT(F, x):
  y = 0 # arbitrarily chosen
  def F_prime(x):
    F(x)
    return y
  return P(F_prime, x, y)
```

We modify F to create F that runs $F(x)$ and if $F(x)$ halts it outputs an arbitrarily chosen output y. We then call $P(f,x,y)$ and if it returns true then $F(x)$ halts and if it returns false then $F(x)$ must not halt. Therefore we have solved the halting problem. This is a contradiction because the halting problem is uncomputable. Therefore the program P cannot exist.

(b) If P exists, we can solve the halting problem. We show this by constructing HALT(F,x) where F is a program and x is the input. We will use P as a subroutine to derive a contradiction.

```
def HALT(F, x):
  def F_prime(ignore):
    return F(x)
  return P(F_prime)
```

We construct a function F which ignores its input and simply runs $F(x)$. We then call $P(F)$. If $P(F')$ returns true then $F(x)$ must have halted, otherwise $P(F')$ will have returned false.

Therefore, we have solved the halting problem. This is a contradiction because the halting problem is uncomputable. Therefore, the program P cannot exist.

(c) If P exists, we can solve the halting problem. We show this by constructing `HALT`(F, x) where F is a program and x is the input. We will use P as a subroutine to derive a contradiction.

```
def HALT(F, x):
  def F_prime(y):
    F(x)
    while x != y:
      pass
    return

  def G(y):
    while x != y:
      pass
    return
  return P(F_prime, G)
```

We construct functions G and F'. Both functions loop forever unless the input is x. Additionally, F' runs $F(x)$ and so only halts if $F(x)$ halts. We then call $P(F', G)$, and if the answer is true, then the the F halts on x. Otherwise, it does not halt on x. Therefore, we have solved the halting problem. This is a contradiction because the halting problem is uncomputable. Therefore, the program P cannot exist.