Exercises List

Yuanming Yu
yyming [at] gmail [dot] com

December 5, 2011

1. Is the graph depicted in Fig 2 Eulerian? Either give an Eulerian cycle or justify why it doesn't exist.

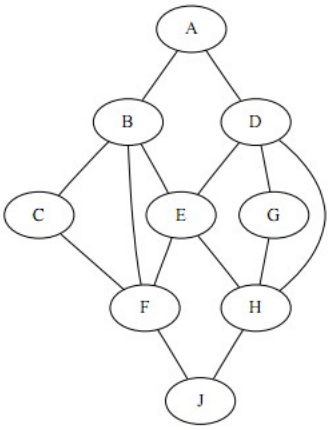


Figure 1: Can you draw this in one stroke?

Solution: Yes. B – C – F – J – H – G – D – A – B – F – E – D – H – E – B.

2. A **leaf** is a vertex with exactly one neighbor. Every tree G with more than one vertex has at least two leaves.

Proof. Let G be an arbitrary connected acyclic graph with more than one vertex.

Because G is connected and has more than one vertex, every vertex has degree at least 1.

Let $v_0, v_1, \ldots, v_n$ be a maximal path in G, that is, a path that cannot be made longer by adding a vertex to either end. Because the path is maximal, it must visit every neighbor of v_n. If v_n is adjacent to v_i for any $i < n-1$, then $v_i, v_{i+1}, \ldots, v_n, v_i$ is a cycle in G. Because G is acyclic, no such cycle exists.

Thus, v_n is adjacent to v_{n-1} and nothing else; in other words, v_n is a leaf. By a similar argument, v_0 is a leaf. □

Other facts about trees that you should prove

- Every connected graph has at least $|V| - 1$ edges.
- Every acyclic graph has at most $|V| - 1$ edges.
- Any connected graph with (at most) $|V| - 1$ edges is a tree.
- Any acyclic graph with (at least) $|V| - 1$ edges is a tree.
- Any minimally connected graph is a tree (We've proved it in later exercise).
- Any maximally acyclic graph is a tree. ("Maximally acyclic" means adding any edge creates a cycle.)
- A graph is a tree if and only if there is a unique path from any vertex to any other vertex.
- Every tree containing a vertex of degree Δ has at least Δ leaves.
- In any tree, a strict majority of the vertices have degree at most 2.

3. **Tree Characterization** Let T be a connected simple graph of order n. Then T is a tree iff the size of T is $n - 1$.

Proof. By definition, the order of a tree is how many nodes it has, and its size is how many edges it has.

Necessary Condition

Suppose T is a tree with n nodes. We need to show that T has $n - 1$ edges.

Proof by induction:

Let T_n be a tree with n nodes.

For all $n \in \mathcal{N}^*$, let $P(n)$ be the proposition that a tree with n nodes has $n - 1$ edges.

Basis for the Induction:

$P(1)$ says that a tree with 1 vertex has no edges.

It is clear that T_1 is N_1, the edgeless graph, which has 1 node and no edges.

So $P(1)$ is (trivially) true.

This is our basis for the induction.

Induction Hypothesis:

Now we need to show that, if $P(k)$ is true, where $k \geq 1$, then it logically follows that $P(k+1)$ is true.

So this is our induction hypothesis: :Any tree with k nodes has $k - 1$ edges.

Then we need to show: Any tree with $k + 1$ nodes has k edges.

Induction Step:

Let T_{k+1} be any tree with $k + 1$ nodes.

Take any node v of T_{k+1} of degree 1. Such a node exists from Tree has Degree One Nodes.

Let us consider T_k, the subgraph of T_{k+1} created by removing v and the edge connecting it to the rest of the graph.

By Subgraph of Tree, T_k is itself a tree.

The order of T_k is k, and it has one less edge than T_{k+1} by definition.

By the induction hypothesis, T_k has $k - 1$ edges.

So T_{k+1} must have k edges.

So $P(k) \implies P(k+1)$ and the result follows by the Principle of Mathematical Induction.

Alternative Induction Step

Let T_{k+1} be any tree with $k + 1$ nodes.

Remove any edge e of T.

As T_{k+1} has no circuits, e must be a bridge, from Condition for an Edge to be a Bridge.

So removing e disconnects T_{k+1} into two trees T_1 and T_2, with k_1 and k_2 nodes, where $k_1 + k_2 = k + 1$.

By the induction hypothesis, T_1 and T_2 have $k_1 - 1$ and $k_2 - 1$ edges.

Putting the edge e back again, we see that T_{k+1} has $(k_1 - 1) + (k_2 - 1) + 1 = k$ edges.

So $P(k) \implies P(k+1)$ and the result follows by the Principle of Strong Induction.

Therefore a tree with n nodes has $n - 1$ edges.

Sufficient Condition

Suppose T is a connected simple graph of order n with $n - 1$ edges.

We need to show that T is a tree.

Suppose that T is not a tree. Then it contains a circuit.

It follows from Condition for an Edge to be a Bridge that there is at least one edge in T which is not a bridge.

So we can remove this edge and obtain a graph T' which is connected and has n nodes and $n - 2$ edges.

Let us try and construct a connected graph with n nodes and $n - 2$ edges.

We start with the edgeless graph N_n, and add edges till the graph is connected.

We pick any two vertices of N_n, label them u_1 and u_2 for convenience, and use one edge to connect them, labelling that edge e_1.

We pick any other vertex, label it u_3, and use one edge to connect it to either u_1 or u_2, labelling that edge e_2.

We pick any other vertex, label it u_4, and use one edge to connect it to either u_1, u_2 or u_3, labelling that edge e_3.

We continue in this way, until we pick a vertex, label it u_{n-1}, and use one edge to connect it to either $u_1, u_2, \ldots, u_{n-2}$, labelling that edge e_{n-2}.

That's the last of our edges, and we still haven't connected the last vertex.

Therefore a graph with n vertices and $n-2$ edges that such a graph can "not" be connected.

Therefore we can not remove any edge from T without leaving it disconnected.

Therefore all the edges in T are bridges.

Hence T can contain no circuits and so must be a tree.

□

4. Hogwarts' annual prom is coming. A dance is conducted by one boy and one girl who know one another. Magically, it so happens that each boy knows exactly d girls, and each girl knows exactly d boys (after all this is a renowned school of witchcraft and wizardry).

   (a) Prove that the edges of a d-regular bipartite graph can be partitioned into d matchings.

   (b) Conclude that there exists an arrangement of dancing such that after d songs (hence d dances), every boy has danced with every girl he knows.

   Solution

   (a) **Lemma.** *A d-regular graph has a perfect matching.*

   Proof. Slide 43 of Lecture 16 (Using Hall's Theorem). □

   Now we prove the main theorem:

   Proof. We prove by induction on d. The base case of $d = 1$ is a graph which is a perfect matching itself. Now assume the edges of a $(d-1)$-regular graph can be partitioned into $d-1$ matchings, for some $d > 1$. For a d-regular graph G, we know that it has a perfect matching M. Observe that $G \backslash M$ is a $(d-1)$-regular graph, by induction hypothesis its edges can be partitioned into $d-1$ matchings. Therefore the edges of G can be partitioned in to d matchings. We conclude that for any $d \geq 1$, the edges of a d-regular graph can be partitioned into d matchings. □

   (b) If we draw a graph G of the acquaintance between boys and girls at Hogwarts, then G will be a d-regular bipartite graph. A matching in G gives us an arrangement to pair up boys and girls to dance in the ballroom. By (a), we can partition the edges of G into d matchings $M_1, M_2, \ldots, M_d$, which gives us an arrangement to let everyboy dance with all d girls he knows: for the i-th song, we let the boys and girls paired up by perfect matching M_i dance.

5. **(k stroke graphs)**: For a connected graph, if there are $2k$ odd degree vertices, then the graph can be drawn in k strokes.

   Proof. If $k > 1$, connects two odd degree vertices u and v, to make a new problem:

   > For a a connected graph, if there are $2(k-1)$ odd degree vertices, then the graph can be drawn in (k-1) strokes.

   The new problem implys the old one because passing through the newly edge (u, v) can be regarded as making a new stroke starting at v (while the previous one ending at u).

   Hence our problem can reduce to

   > For a connected graph, if there are 2 odd degree vertices, then the graph can be drawn in 1 strokes.

   This is the Euler Path setting, the problem is certainly solvable, so does our original one. □

6. **(tree)**: prove a graph is a tree if and only if it is "minimally" connected, i.e. removing any edge would disconnect the graph.

   Proof. tree → 'minimally' connected: by definition, a graph is a tree if and only if there is a unique path between any pair of vertices. If we remove (u, v), then u and v will be disconnected because (u, v) is the unique path. Thus a tree is 'minimally' connected.

   'minimally' connected → tree: to prove that minimally connected implies a tree, we can prove that for a connected graph having a cycle is not minimally connected (since delete arbitrary one edge on cycle the graph remains connected). The **contrapositive** is that for a connected graph, minimally connected implies no cycles. Then we are done because one definition for tree is that: tree is a connected graph with no cycle.

   An alternative proof that minimally connected implies a tree, pick any edge e and delete it. Each connected component is minimally connected otherwise, reduce edges to make every component minimally connected, and then add e back to make the graph connected again, result into a smaller but connected graph, contradiction. Then by induction each connected component is a tree(easy check), and thus the original graph is a tree itself.

   Thus finish our proof.

   □

7. Mr. and Mrs. Smith held a party at home and n couples came. A few handshakes took place. Mr Smith observed that:

   - No couples shook hands;
   - Nor did anyone shake hands with himself or herself;
   - Nobody shook hands with the same person more than once;
   - Number of handshakes of others (the n couples and Mrs Smith) are distinct.

   (a) For $n = 3$ and $n = 4$, calculate the number of handshakes Mrs. Smith and Mr. Smith had.

   (b) What is the number of handshakes Mr. and Mrs. Smith had for the general case? Prove your answers.

   Proof. We prove the general case by induction on n.

   Base case: when $n = 1$, there are two guests and it's easy to verify the only solution is one guest having no handshakes and the other guest shook hands with both the Smiths. And the Smiths both had $n = 1$ handshake(s). Suppose when $n - 1$ couples come and Mr. Smith finds everybody else has different number of handshakes, then the Smiths both had $n - 1$ handshakes, for some $n > 1$.

   For notational convenience we denote by P_i the guest (or Mrs. Smith) with i handshakes. P_{2n} shook hands with everybody except his spouse, hence P_0 has to be his spouse. Now we show this couple cannot be the Smiths: We observe that Mr. Smith cannot have 0 handshakes, or P_{2n} can at most have $2n - 1$ handshakes. Neither can he have $2n$ handshakes, or P_1 will have at least 2 handshakes. Now if we take this couple out of the picture, everybody else's handshake count drops by one. The number of handshakes among the Smiths and the $n - 1$ couples are $0, 1, 2, \ldots, 2n - 2$ and that of Mr. Smith's. By induction hypothesis, the Smiths both had $n - 1$ handshakes with those $n - 1$ couples. Adding back the 1 handshake they each had with the P_{2n}'s, the Smiths both had n handshakes. Therefore for any $n \geq 1$, the Smiths both have n handshakes. ☐

8. Determine if each pair of graphs that follows are isomorphic.

   (a) See Fig 2

   (b) See Fig 3

   Solution:

   (a) YES. The mapping is follow: A(2), B(3), C(4), D(1), E(5), F(6)

   (b) YES. The mapping is follow: A(A), B(C), C(E), D(B), E(D)

9. Are the following sequences valid degree sequences of simple graphs? For each sequence, do one of the following:

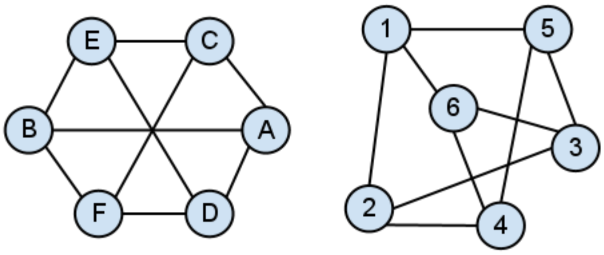


Figure 2: isomorphic graphs determine

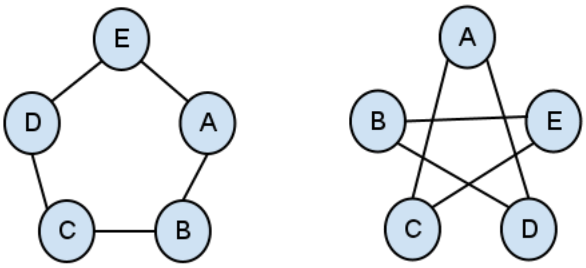


Figure 3: isomorphic graphs determine

- Construct a graph with the given degree sequence, or
- Show that the sequence cannot be a degree sequence of any simple graph

(a) (3,3,3,1)
(b) (4,4,4,2,2)
(c) (4,3,2,2,1)
(d) (3,3,3,3,3,3)

Solution:

(a) (3,3,3,1)
This sequence is not graphical. Each of the three vertices of degree 3 can at most have two edges from other degree-3 vertices, but then the degree 1 vertex can offer only one more edge.

(b) (4,4,4,2,2)
Similarly, this sequence is not graphical.

(c) (4,3,2,2,1) This sequence corresponds uniquely to a simple graph as shown in

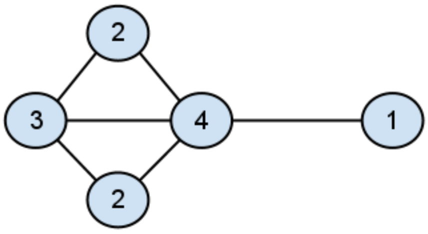


Figure 4: Graph with degree sequence (4,3,2,2,1)

Figure 4

(d) (3,3,3,3,3,3)

This sequence is graphical. The graph in Figure 2 has such a degree sequence.

10. Show that the Gale-Shapley algorithm is men-optimal, and also women-pessimal.

Proof. **Claim:** If a woman w rejects a man m during the execution of the Gale-Shapley algorithm, then (m, w) cannot be valid partners.

We prove the claim by induction. In the first iteration, if w rejects a man m, then there is a man m^* who also proposes to w and m^* is higher on w's list than m. Suppose, by way of contradiction, that (m, w) are matched in a stable matching M. Since w is the first on m^* list(as m^* proposed to w in the first iteration), m^* prefers w to his partner in M. This implies that (m^*, w) is an unstable pair in M, which constradicts that M is a stable matching.

Now assume the claim is true for the k-th iteration, we prove that it is also true for the $k+1$-th iteration. The argument is almost the same as above. If w rejects a man m in the $k+1$-th iteration, then there is a man m^* who also proposes to w in the $k+1$-th iteration and m^* is higher on w's list than m. Suppose (m, w) are matched in a table matching M. By induction hypothesis, the women that have rejected m^*((in or before the k-th iteration) cannot be valid partners for m^*. Since m^* proposed in a non-increasing order, m^* prefers w to his partner in M. This implies that (m^*, w) is an unstable pair in M, which constradicts that M is a stable matching.

Now goes our main proof: Gale-Shapley algorithm is men-optimal, and also women-pessimal.

Since a man proposes in a non-increasing order, by Claim, the first woman who does not reject him is his best valid partner. Similarly, since the sequence of proposals that a woman is holding is non-decreasing, by Claim, the first man she does not reject is her worst valid partner. □

11. Show an example that women can lie to get a better partner.

    Someone asked how cheating is possible in the stable marriage problem. Here is an example of 3 men (a,b,c) and 3 women (1,2,3). Suppose the algorithm is the Gale-Shapley algorithm and is men-optimal. Consider the following "true" preference lists.

    man a prefers $2 > 1 > 3$

    man b prefers $1 > 2 > 3$

    man c prefers $2 > 3 > 1$

    woman 1 prefers $a > b > c$

    woman 2 prefers $b > a > c$

    woman 3 prefers $c > b > a$

    Then the Gale-Shapley (men-optimal) algorithm would find the matching (a,2),(b,1),(c,3).

    Note that woman 2 only gets her second choice if she uses her true preference list. In fact, woman can get her first choice (man b) if she uses the following "faked" list.

    woman 2 prefers $b > c > a$

    Now, the Gale-Shapley (men-optimal) algorithm would find the matching (a,1),(b,2),(c,3). Try it.

12. (a) Think about Stable matching when there are more applicants than positions scenario.
    (b) Generalize the definition of unstable pairs.
    (c) Use the Gale-Shapley result to show that there is a stable matching in this more general setting.

    Solution:

    (a) Adding dummy positions with number equal to number of applicant minus number of position, and put those positions at the end of every applicant's preference list;
    (b) the definition of unstable pair is the same as the old one, besides that for every dummy positions, all applicants are the same.
    (c) For dummy position, just set an arbitrary preference list order to every applicants.

        Now that the problem can be regarded as a Stable Marriage Problem, by Gale-Shapley there is a stable matching.

Since for every dummy position, there is no difference among applicants, so the specific preference version has no unstable pair imply that the general setting has no unstable pair, thus it is also a stable matching.

13. Show that a bipartite graph has a matching of size $n - k$ if and only if there is no subset S of one side with $|N(S)| + k < |S|$, where n is the number of nodes on one side and so the graph has $2n$ nodes.

    Remark: This is a generalization of Hall's theorem.

    Solution:

    - Reduce the problem to perfect matching by adding dummy nodes $x_1, x_2, \ldots, x_k$ on the left side and $y_1, y_2, \ldots, y_k$ on the right side.
    - dummy nodes connect to every node in the opposite side, i.e. for $u \in left[G]$, adding (u, y_i), and for $v \in right[G]$, adding (x_i, v), and adding (x_i, y_j).
    - Then there is a perfect matching in the new graph if and only if there is a matching of size $n - k$ in the old graph: 1) the old graph has a matching of size $n-k$, clearly the new one has a matching of size $n+k$ (for the $2k$ dummy nodes can make pair with any node); 2) the new one has of size $n + k$, then delete those pairs has dummy vertices will affect at most $2k$ pairs, thus we have a matching of size $n - k$ in the old graph.

The problem we want to prove is that: a bipartite graph has a matching of size $n - k$ if and only if there is no subset S of one side with $|N(S)| + k < |S|$, which is equivalent to: a bipartite graph has no matching of size $n - k$ if and only if there is a subset S of one side with $|N(S)| + k < |S|$ (contrapositive and the property of iff).

One direction is easier: if there is a set S with $|N(S)| + k < S$ then there is no matching of size $n - k$. In fact, if there is a set S with $|N(S)| + k < S$, then the whole graph's matching size is at most $|N(S)| + |V - S|$ because at most $|N(S)|$ vertices of S can be matched, and this is at most $|S| - k - 1 + |V - S| = n - k - 1$.

The other direction is harder. If the graph has no matching of size $n - k$, then there is a subset S of one side with $|N(S)| + k < |S|$.

We use the fact that there is a matching of size $n-k$ in the old graph if and only if there is a perfect matching in the new graph. By Hall's theorem, there is no perfect matching in the new graph if and only if there is a set S' with $|N(S')| < |S'|$ in the new graph.

Since the old graph has no matching of size $n - k$, we know that there is set S' with $|N(S')| < |S'|$ in the new graph.

Note that the dummy nodes cannot be in S', because they connect to everyone in the opposite side, which would make $|N(S')| = n + k$ and further the inequality do not hold.

Also note that the dummy nodes on the other side must be in $N(S')$, because they connect to everyone to the opposite side, and S' is not empty thus there is one node in it connects to all these k dummy nodes.

Delete the dummy nodes on both sides, we should have $S = S'$(since we proved above that there is no dummy nodes in S') and

$$N(S) = N(S')\backslash\{\text{all } k \text{ dummy nodes on that side}\}$$

so that $|N(S)| + k = |N(S')|$. Then we see that $|N(S)| + k < |S|$, and we are done.

In sum, a bipartite graph has a matching of size $n - k$ if and only if there is no subset S of one side with $|N(S)| + k < |S|$.

14. An example that degree upper bounds could be any positive number.

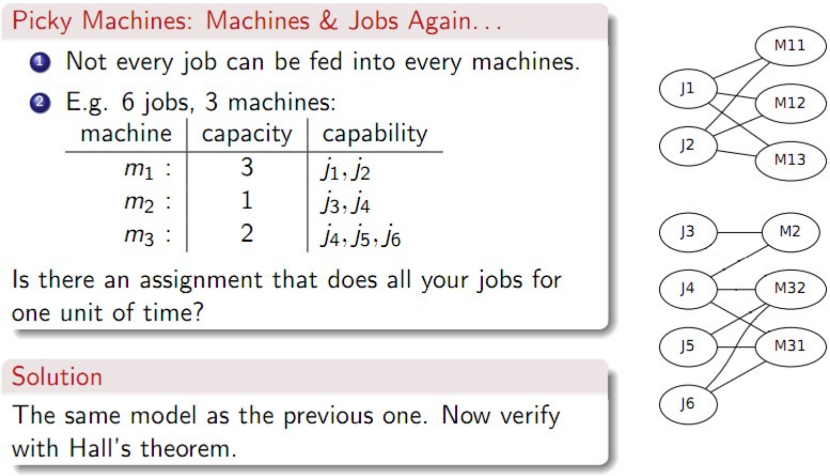
    - n jobs (j_i) each taking one unit of time.
    - m machines, machine m_i can run c_i jobs.
    - And define happily $\sum_{i=1}^{m} c_i$ = n

    (a) Is it possible to assign jobs to machines so that you only need to wait one unit of time and finish them all?

    (b) Model this as a bipartite matching problem. Show the correspondence (how you can construct an instance of bipartite matching problem given the sepc of jobs and machines).

    Solution:

    - a job $j_i \rightarrow j_i$ on the left
    - a machine m_i with capacity $c_i \rightarrow c_i$ vertices $\{m_{i,j} | j = 1, 2, \ldots, c_i\}$ on the right.
    - each vertex on the right is connected to all vertices on the left
    - An assignment equivalent to a (prefect) matching



Picky Machines: Machines & Jobs Again...

1. Not every job can be fed into every machines.
2. E.g. 6 jobs, 3 machines:

machine	capacity	capability
m_1 :	3	j_1, j_2
m_2 :	1	j_3, j_4
m_3 :	2	j_4, j_5, j_6

Is there an assignment that does all your jobs for one unit of time?

Solution

The same model as the previous one. Now verify with Hall's theorem.

Figure 5: Verify