Discrete Mathematics	DATE: 10.08.2018

Assignmemt 3 Part II

Instructor: Sourav Chakraborty | *Scribe: Soumee Guha*

8. Prove that a graph is bipartite if and only if the graph has no odd cycle.

A bipartite graph has no odd cycle.

Let G = (V,E) be a bipartite graph with bipartition X, Y. Let u be a vertex in non-trivial component H. For each v ∈ V(H), let f(v) be the minimum length of u,v path. Since H is connected, f(v) is defined for each v ∈ V(H).

Let X = { v ∈ V(H): f(v) is even } and Y = { v ∈ V(H): f(v) is odd }. An edge within X or Y would create an odd walk using the shortest path. In any bipartite graph, every walk alternates between the two sets of a bipartition, so every return to the original partite set happens after an even number of steps. Hence, a bipartite graph cannot contain an odd cycle.

A graph with no odd cycle is bipartite.

To prove this, let us consider the partitions X and Y mentioned in the previous case. Let two vertices in X or Y be adjacent. However, according to our definition, this cannot happen. Also, an odd cycle can exist only if two vertices in X or Y are adjacent. Hence, no odd cycle can exist here as every walk alternates between the two sets. Hence, the graph is bipartite.

Hence it is proved that a graph is bipartite if and only if the graph has no odd cycle.

9. If G is a graph such that all vertices has degree more that 2 then G has a cycle.

Let P be the largest path in G = (V, E) in which no vertex is repeated. Let P connect vertices v_i, i ∈ {0,1 .. t}. Let every vertex in G have degree at least 2. Let us consider the vertex v_t. Since the degree of all the vertices is at least 2, v_t should have at least 2 neighbours. v_{t-1} is a neighbour of v_t. Since P is the longest path, it cannot be extended. The other neighbour of v_t must be a vertex in P. Hence G must contain a cycle.

10. If G is a graph such that all the vertices has even degree then prove that G can be written as a union of edge-disjoint cycles.

We shall induct on the number of edges.

Base Case: Let us consider a graph having 3 edges. This is because if |E| < 3, we cannot have a cycle. A connected graph in which each vertex has even degree and 3 edges must be a triangle because

$$2.3 = \sum_{v \in V} d(v)$$

$$\Rightarrow \sum_{v \in V} d(v) = 6$$

$$6 \geq 2|V|$$

which gives | V | = 3 because any graph on | V | < 3 cannot have a cycle. Hence, P(3) holds.

Induction Hypothesis: Let P(k) be true.

Inductive Step: Let G be a connected graph on k + 1 edges. Now, the degree of each vertex is even and degree $\geq$ 2, G has a cycle. Let us denote this cycle as C and consider G \ E(c). Since C has at least 3 edges, G \ E(C) has edges $\leq$ k. Now, if a vertex v of G appears in G m times, then after deletion of E(C) from G, in G \ E(c), degree of v will be d(v) - 2m. Thus, each vertex of G \ E(c) still has even degree (since d(v) is even for each v). Each connected component of G \ E(c) has all vertices even degree and number of edges $\leq$k. By induction hypothesis, it can be written as union of edge disjoint cycles $\cup_i C_i^j$ (where index j is for the jth connected component of G \ E(c)). Now for each (i,j), C_i^j is edge disjoint with C since C_i^j belongs in G \ E(c). So, G can be written as $\cup_j \cup_i C_i^j \cup C$. Thus, C can be written as edge disjoint cycles.

11. A simple undirected graph where any two vertices are connected is called a connected graph. A simple undirected graph which has no cycle is called acyclic. An acyclic connected graph is called a tree.

1. **Prove that every tree has atleast 2 vertex of degree 1.**

   We shall induct on the number of vertices.
   Base Case: Let us consider a tree G = (V, E). In Figure 1, we can see that there are 2 vertices of degree one. So the base case P(2) holds.
   Inductive Hypothesis: Let us consider a tree with k vertices. Let us assume that it has atleast 2 vertices of degree 1, i.e, P(k) holds.
   Inductive Step: Say we have a tree with k + 1 vertices (G). Let the graph G' =$G \setminus v$ be considered. By induction hypothesis, G' has at least 2 vertex of degree 1. Now, v is a leaf and considering the vertex v in G, there will be one more edge in G. Since in the tree G any leaf has degree 1, v is connected to only one vertex. Hence, G will have k + 1 vertices with atleast 2 vertices of degree 1. Therefore, P(k) $\Rightarrow$ P(k+1).

   Hence it is proved that every tree has atleast 2 vertex of degree 1.

2. **Prove that a tree on n vertices has exactly (n-1) edges.**

   We shall induct on the number of vertices.
   Base Case: Let us consider a tree G = (V, E) having 2 vertices as shown in Figure 1. It has 2 vertices with 1 edge. Therefore P(2) holds.

Figure 1: Base case

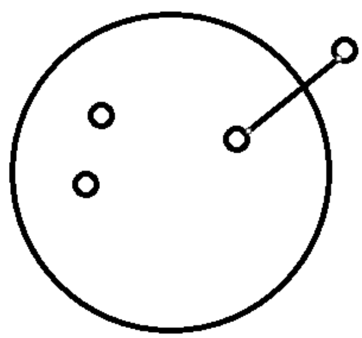

Figure 2: Graph G' = $G \setminus v$

Inductive Hypothesis: Let us consider a tree with k vertices. Let us assume that it has exactly k - 1 edges, i.e, P(k) holds.
Inductive step: Say we have a tree G with k + 1 vertices. Let the graph G' =$G \setminus \{v\}$ be considered where v is a leaf. By induction hypothesis, G' has k - 1 edges. In a tree, a leaf has degree 1. Therefore, v is connected only one edge and removing v will decrease the number of edges by 1. Therefore, G has (k - 1) + 1, i.e, k edges. Therefore P(k) ⇒ P(k + 1).
Hence it is proved that a tree on n vertices has exactly n-1 edges.

3. **Prove that is a graph G is connected and has exactly (n-1) edges then G is a tree.**

   Let G = (V, E) be a connected graph on n vertices. Let us further assume that G contains atleast one cycle. Let an edge be removed from G such that the resulting graph is connected. Let this process be continued until the resulting graph is acyclic, i.e, a tree. As there is no cycle, the number of edges in the acyclic graph will be n-1. Now, the number of edges in G is greater than the number of edges in H. But G contains n-1 edges. So G is acyclic having n-1 edges. Therefore, it is proved that G is a tree.

4. **A tree where every vertex has degree 3 or 1 (except the root that has degree 2 and degree zero is the tree is of size 1) is called a binary tree. Prove that a binary tree on n vertices has at least $\lfloor n/2 \rfloor$ + 1 number of leaves.**

   We shall prove this by induction. Let us induct on the number of vertices of a graph.
   Base Case: Let us consider a tree comprising 3 nodes. In that case there will be

root node and ($\lfloor n/2 \rfloor$ + 1) = 2 leaf nodes. Hence P(3) holds.
Inductive Hypothesis: Let this hold for k vertices, i.e, let P(k) hold.
Inductive Step: Let us consider a binary tree G having n+1 vertices. Let us remove the root node. By doing so, we shall get two binary trees, G1 and G2 and the two neighbours of the root node in G will be the root nodes in the newly obtained ones. Let v1 and v2 be the neighbours of the root node in G. Let k1 be the number of nodes in the binary tree whose root is v1 (G1) and k2 be the number of vertices in G2. Therefore, k1 + k2 = n. By induction hypothesis, G1 has $\lfloor k1/2 \rfloor$ + 1 leaves and G2 has $\lfloor k2/2 \rfloor$ + 1 leaves. Hence, G has $\lfloor k1/2 \rfloor$ + $\lfloor k2/2 \rfloor$ + 2 leaves.
Case I:
n + 1 is even
⇒ n is odd
⇒ Either k1 is odd or k2 is odd
Let us assume that k1 is odd.
$\lfloor k1/2 \rfloor + \lfloor k2/2 \rfloor + 2$
$= \frac{k1-1}{2} + \frac{k2}{2} + 2$
$= \frac{k1+k2-1}{2} + 2$
$= \frac{k1+k2+1}{2} + 1$
$= \frac{n+1}{2} + 1$
$\geq \lfloor (n+1)/2 \rfloor + 1$
Case II:
n + 1 is odd
⇒ n is even
⇒ k1 and k2 are either odd or even.

- Both k1 and k2 are even
  $\lfloor k1/2 \rfloor + \lfloor k2/2 \rfloor + 2$
  $= \frac{k1+k2}{2} + 2$
  $= \frac{n}{2} + 2$
  $\geq \frac{n+1}{2} + 1$
- Both k1 and k2 are odd
  $\lfloor k1/2 \rfloor + \lfloor k2/2 \rfloor + 2$
  $= \frac{k1-1}{2} + \frac{k2-1}{2} + 2$
  $= \frac{k1+k2}{2} + 1$
  $= \frac{n}{2} + 1$
  $= \lfloor$ (n + 1)/2 $\rfloor$ + 1

Hence by induction, G has $\lfloor$ (n + 1)/2 $\rfloor$ + 1 number of leaves.

12. Following are three wrong statements and their proofs. Find out where the mistake is in each of the three case.

1. **Statement: All cows are the same colour.**

   This statement is proved by induction taking the base case as P(1). In the inductive step it is assumed that in any collection of n cows all n of them are the same colour.

A set n + 1 of cows numbered 1 to n + 1 is considered. By the induction hypothesis cows 1 to n are the same colour and similarly cows 2 to n + 1 are the same colour. But the middle cows 2 to n cant change colour according to who they are grouped with so cows 1 to n + 1 must all be the same colour.

However as the base case is P(1), $\forall$ n $\geq$ 2 the induction will not be valid. Let us consider n = 2. Since the base case is P(1), there will not be any overlap between two cows c1 and c2 as is evident from Figure 3. Here (P(1) is true and $\forall$ n $\geq$ 1 P(k) $\Rightarrow$ P(k+1)) $\Rightarrow$ $\forall$ k, P(k) is true. The inductive step should have been such that $\exists$ at least one cow that falls in both the categories. This can only happen if k+1 > 3. Only then this argument will be valid.

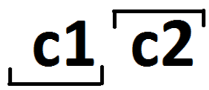

Figure 3: Considering the case n = 2

2. **Statement: For a list of length n, mergesort takes O(n) time.**

   In the proof, P(n) denotes the number of steps taken by merge sort to sort n items.
   Base Case: merge sort on the empty list just returns the empty list. So P(0) = 0.
   Strong induction: Assume P(1), ..., P(n 1) and try to prove P(n). We know that at each step in a recursive mergesort, two approximately half-lists are mergesorted and then zipped up. It is mentioned that mergesorting of each half list takes, by induction, O(n/2) time which is not correct. Here it is assumed that P(k): T(k) = O(k)
   The algorithm has a recurrence relation of M(n) = 2M(n/2) + O(n). The recurrence relation implies that $\exists$ c, s.t T(n) $\leq$ T(n/2) + T(n/2) + cn.
   Now, P(k): T(k) $\leq$ dk
   By strong induction, Π P(i), i $\in$ { 1, .. k } = P(k+1).
   T(k+1)
   $\leq$ T($\frac{k+1}{2}$) + T($\frac{k+1}{2}$) + ck
   $\leq$ d($\frac{k+1}{2}$) + d($\frac{k+1}{2}$) + ck
   $\leq$ d(k+1) + ck
   We need to show that T(k+1) $\leq$ d(k+1). However, this cannot be done as c is a constant. So, the induction is not correct.

3. **Statement: Every graph with more than 3 vertices and minimum degree 2 contains a 3-cycle.**

   The induction has been done on | V(G) |. From Figure 3, it can be seen that the theorem | V(G) | = 3 is true when since any simple graph on three vertices with all vertices of degree $\geq$ 2 must be a cycle of length 3.
   P(k) = $\forall$ G with | V(G) | $\leq$ k and minimum degree $\geq$ 2 has a 3-cycle.
   P(3) $\wedge$ { P(k) $\Rightarrow$ P(k+ 1) $\forall$ k $\geq$ 3 } then, { $\forall$ k , P(k) is true.}

   Inductive step: P(k) $\Rightarrow$ P(k+1) Let graph G be in P(k). G has k vertices and minimum

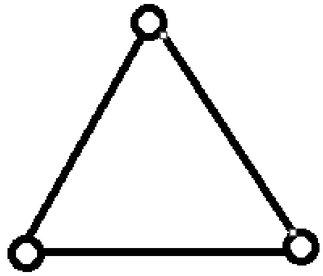

Figure 4: Base Case: P(3)

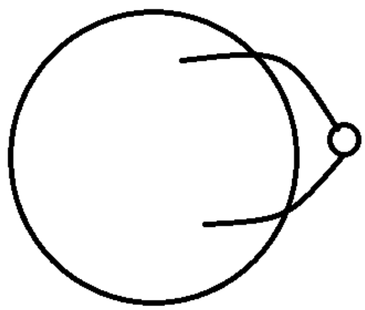

Figure 5: Example Graph

degree $\geq$ k. As $\exists$ a 3-cycle in P(k), $\exists$ a 3-cycle in P(k+1).
Now let us consider Figure 6.Even though there is a 3-cycle in P(3), no such cycle exists in P(4) which clearly proves that the proof is not correct. In case of graphs, we

Figure 6: P(3) and P(k+1)

need to move from a higher level to lower level and we should never try to proceed in the reverse order.