ACM Summer School 2019

Graph Theory and Graph Algorithms

Problem sheet

Date: 17.06.2019 **Week:** 1 **Day:** 1

1. Prove that in an undirected graph there are an even number of vertices of odd degree.

2. Prove that if u is a vertex of odd degree in a graph, then there exists a path from u to another vertex v of the graph where v also has odd degree.

3. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_)$ are isomorphic if there is a bijection $\phi : V_1 \rightarrow V_2$ such that $\{u, v\} \in E_1$ if and only if $\{\phi(u), \phi(v)\} \in E_2$. Clearly, if two graphs are isomorphic, the degree sequences of the two graphs are identical. Construct two non-isomophic graphs which have the same degree sequence.
   Reading Exercise: `http://compalg.inf.elte.hu/~tony/Oktatas/TDK/FINAL/Chap%202.PDF`.

4. (a) In IPL (Indian premier cricket league) 2018, at the end of May 18, 2018, each of the eight teams has completed 13 games each. Suppose there were totally 9 teams, is it possible to design fixtures so that each team has completed 13 games each at the end of some day?
   (b) On May 20 th the last two matches of the league were played and at the end of the day, each of the 8 teams has played against each other team twice. My son says that the matches played on 20th were 53rd and 54th matches of the tournament. Is he correct? Explain.
   (c) Can you design a fixture so that each of the 8 teams has played different number of games? I.e. no two teams have completed the same number of matches. Recall that each team plays against every other team twice and at the beginning of day 1, each team has played 0 matches. (Hint: start with, say 4 teams first).
   (d) Can you design a fixture so that each of the 8 teams has played different number of games at the end of some day, when each team is to play against every other team exactly once?

5. Consider an $n \times n$ chess board and consider the graphs obtained based on the moves of the Rook, Bishop, Horse, King, Queen, and the Pawn. Classify the graphs based on their properties: Directed, undirected, connected, Eulerian, Hamiltonian, and other Graph theoretic properties.

6. How do you determine whether a graph is connected? What is the complexity of your algorithm?

7. How do you determine whether a given graph is bipartite? What is the complexity of your algorithm?

8. How do you determine whether a given graph has a cycle in $O(n)$ time (regardless of the number of edges in the graph)?

9. A bridge in a connected undirected graph is an edge whose removal disconnects the graph.
   (a) Prove than an edge is a bridge if and only if there is no cycle containing that edge.
   (b) Give an algorithm to find all bridges in a connected undirected graph. What is the running time of the algorithm?

10. A directed graph G has an Euler tour if there is a directed cycle that visits every edge exactly once (though vertices can be repeated several times).
    (a) Show that a directed graph has an Euler tour if and only if for every vertex, its indegree and outdegree are the same.
    (b) Give an $O(m + n)$ algorithm to find an Euler tour in a directed graph where every vertex has indegree and outdegree the same.

11. A cut vertex x in an undirected graph G is a vertex such that $G \setminus x$ is disconnected.

    (a) Given an undirected graph on n vertices and m edges, how will you find all the cut vertices? How much time does your algorithm take?

    (b) Suppose that the graph has no cut vertex, and consider the DFS tree of the graph. Show that the root can not have more than one child in the tree.