CS 350 Theory of Computation

Assignment 1 (8 marks)

Question 1 (2 marks)

Given an undirected graph \( G = (V, E) \), the breadth-first-search starting at \( v \in V \) (\( \text{bfs}(v) \) for short) is to generate a shortest path tree starting at vertex \( v \in V \). The diameter of \( G \) is the longest of all shortest paths \( \delta(u, v), u, v \in V \).

When \( G \) is a tree, the following algorithm is proposed to compute the diameter of \( G \).

1. Run \( \text{bfs}(w), w \in V \) and compute the vertex \( x \in V \) furthest from \( w \).
2. Run \( \text{bfs}(x) \) and compute the vertex \( y \in V \) furthest from \( x \).
3. Return \( \delta(x, y) \) as the diameter of \( G \).

Prove that this algorithm is correct; i.e., \( \delta(x, y) \) is in fact the longest among all the shortest paths between \( u, v \in V \).

Question 2 (2 marks)

Given a convex polygon \( C(n) \) with \( n \) vertices, prove that one can always decompose \( C(n) \) into triangles using \( n - 3 \) diagonals (a diagonal is a line segment connecting two vertices of \( C(n) \)).

Question 3 (2 marks)

Show that in any simple graph there is a path from any vertex of odd degree to some other vertex of odd degree.

Question 4 (2 marks)

A fully binary tree \( T \) is a tree such that all internal nodes have two children. Prove that a fully binary tree with \( n \) internal nodes in total has \( 2n + 1 \) nodes.

Date Due: before the end of class on Thursday, February 3, 2005. Late assignment will lose 2 marks for each overdue day.