CS338 HW1

Brock Ellefson

February 6, 2017

1

$\frac{1}{3}k(4k^2 - 1) = \frac{1}{3}(2k-1)(2k+1)$

$P(k+1) = 1^2 + 3^2 + 5^2 + ...(2k-1)^2 = \frac{1}{3}(2k-1)(2k+1)$

$= 1^2 + 3^2 + 5^2 + ... + (2k-1)^2 + (2k+1)^2 = \frac{1}{3}(2k-1)(2k+1) + (2k+1)^2$

$= \frac{2k+1}{3}[k(2k-1) + 3(2k+1)]$

$= \frac{2k+1}{3}[2k^2 - k + 6k + 3]$

$= \frac{1}{3}(k+1) + (2(k+1)-1)(2(k+1)+1)$

2

This is false

|V|=2, |E| =7,

|F| = 2 + 7 - 2 = 7

7 > 4 = 2n

3

There are two cases:

1: Simple graph is not a cyclic graph therefore u!=v

2: Simple graph is a cyclic graph therefore u=v

Case 1:

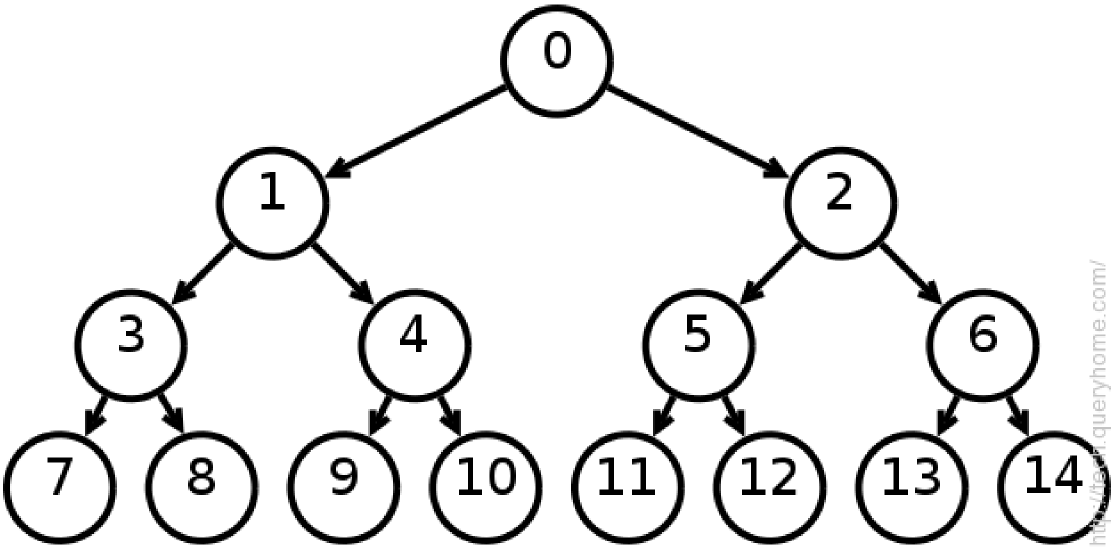
Assumption: A path starting with an odd degree ends with an even degree.

The sum of degrees is 2(Edges)

Therefore, there must be a minimum of one vertex with an odd degree that is a connected component. Being a connected component means that its a pair with each of the vertices.

Therefore, there is always a path from one vertex with an odd degree to another via the connected component

Case 2: If u is an odd degree, then by the definition of a cyclic graph v must be odd as well



4

Assume that that a binary tree B has n internal nodes and i leaves.

Before proving

Base Case: B(0): This binary tree only has a root. Since this node has no children, it is, by definition, a leaf. 2(0) + 1 = 1 node.

Induction: B(n): Assuming this tree has $n \geq 1$ we then know that if will split n into two subtrees n will have a left subtree and a right subtree . This two subtrees will have such a size that l(left subtree) + r(right subtree) + 1(root) = n. The number of leaves on l will be l+1 and the number of leaves on r will be r+1 (we find this by preforming this on B(l) and B(r) and so on until we are only left with the minimum sized subtrees)

Therefore this means i = l + r + 2.

Therefore i = n+1.

i + n = total nodes

(n + 1) + n = total nodes

2n + 1 = total nodes

5

A tree, by definition will only have one path between two nodes.

Therefore, the diameter is that said path between the two nodes

Calling this algorithm on an internal node will return a leaf node, and calling this algorithm on a leaf node will return the node the most far away from the leaf.

This algorithm is correct