Finding Adjacent Nodes

Problem published by Chris santamaria

On the edabit.com Website [1]

Paper by G6W9

**Preface**

I want to clearly state that this paper is of a rather silly nature, as its solution can be condensed into two lines of code, however the structure and formulation of the challenge can be quiet overwhelming, which is the exact reason why this paper exists.

Additionally, this paper will delve deeper into the nature of adjacency matrices and their applications in various fields. The adjacency matrix, a fundamental concept in graph theory, provides a compact representation of the connections within a network. By exploring the properties and uses of adjacency matrices, I aim to shed light on their significance in understanding the structure and behavior of complex systems.

**Tabel of Contents**

[**1** **Introduction** 4](#_Toc170921581)

[**1.1** **Introduction** 4](#_Toc170921582)

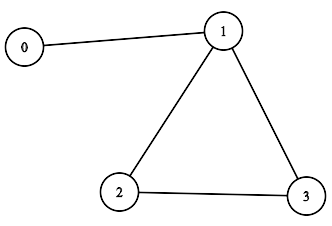
[**1.2** 4](#_Toc170921583)

[**2** **Source** 6](#_Toc170921584)

# **Introduction**

## **Introduction**

In this paper we are going to explore the Problem of finding adjacent nodes in a graph connected by edges on a 2-dimensional plane. Two nodes count as adjacent as long as they share at least one line. Every node as at least one other adjacent node. The maximum number of nodes is limited to 25.000. The longest the program is allowed to take is one hundred milliseconds.



## **Input format and contend.**

For two nodes in a graph to be considered adjacent to one another, there must be an edge between them. In the example given above, nodes 0 and 1 are adjacent, but nodes 0 and 2 are not.

We can encode graphs using an adjacency matrix. An adjacency matrix for a graph with "n" nodes is an "n \* n" matrix where the entry at row "i" and column "j" is a 0 if nodes "i" and "j" are not adjacent, and 1 if nodes "i" and "j" are adjacent.

For the example above, the adjacency matrix would be as follows:

matrix = [

  [ 0, 1, 0, 0 ],

  [ 1, 0, 1, 1 ],

  [ 0, 1, 0, 1 ],

  [ 0, 1, 1, 0 ]

]

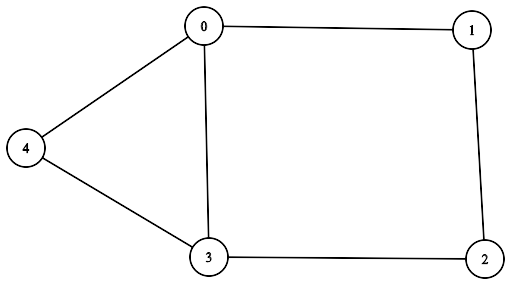
A one indicates that a connection is present and a zero indicates a connection isn’t present.

Here is how the above model might be written out:

* On the first row, node 0 does not connect to itself, but it does connect to node 1. It does not connect to node 2 or node 3. The row is written as 0, 1, 0, 0.
* On the second row, node 1 connects to node 0, node 2 and node 3, but it does not connect to itself. The row is written as 1, 0, 1, 1.
* On the third row, node 2 does not connect to node 0, but it does connect to node 1, does not connect to itself, and it does connect to node 3. The row is written as 0, 1, 0, 1
* On the fourth row, node 3 does not connect to node 0, but it does connect to node 1 and node 2. It does not connect to itself. The row is written as 0, 1, 1, 0.

Your task is to determine if two nodes are adjacent in an undirected graph when given the adjacency matrix and the two nodes.

**Example:**



**Corresponding Matrix:**

matrix = [

  [ 0, 1, 0, 1, 1 ],

  [ 1, 0, 1, 0, 0 ],

  [ 0, 1, 0, 1, 0 ],

  [ 1, 0, 1, 0, 1 ],

  [ 1, 0, 0, 1, 0 ]

]

# **Adjacency matrix**

## **Mathematical Properties**

## **Types of Graphs**

## **Algorithms and Computations**

# **Source**

[1]: https://edabit.com/challenge/3DAkZHv2LZjgqWbvW