

There are 1 questions for a total of 18 points.

- (18) 1. (*Strong Connectivity in Directed Graphs*): This is the programming assignment corresponding to Problem 6, part (b) of HW-2.

Given a directed graph $G = (V, E)$ that is not strongly connected, determine if there exists a pair of vertices $u, v \in V$ such that the graph $G' = (V, E \cup \{(u, v)\})$ becomes strongly connected. In other words, you need to find whether adding a single directed edge from vertex u to vertex v can make the entire graph strongly connected.

You have to implement your algorithm for the above problem. Your program should take input from a file named `input.txt` and should write the output in a file named `output.txt`. Your program should produce an output within 15 seconds for this assignment. The format for input and output files is as follows.

INPUT: The first line of the input file gives the number of vertices n (assume $n \leq 1000$) in the input graph. n lines follow this where the i^{th} line contains the adjacency list of the i^{th} vertex. Below is an example of an input file.

```
6
v1->v2
v2->v1->v3
v3->v1
v4->v5->v1
v5->v6
v6->v4
```

OUTPUT: The first line of the output should contain the strings either **yes** or **no**. Your algorithm should output **yes** if such an edge exists, followed by an edge in the next line (output any one of them if there is more than one such edge). If your answer is **no**, you need not print anything in the second line. For example, consider the output file corresponding to the input file above:

```
yes
v1->v4
```

Note: There are no whitespaces or newline characters at the end of the output file. Having such whitespace might lead to the autograder marking the output as incorrect. There may be multiple possible solutions (edges) in case an edge exists for a given input. In such cases, you output only one such edge and we will verify it.

SUBMISSION INSTRUCTIONS: All your program files should be in a directory named `hw2_prog`. You will be asked to create a zip of this directory and submit this zip file. In this directory, there should be a makefile that will compile your code (read about makefile on the net in case you do not know what it is). After running `make`, the directory should have an executable called `scgraph`. This, when executed, should read the input file (`input.txt`) and write the answer in the output file (`output.txt`).

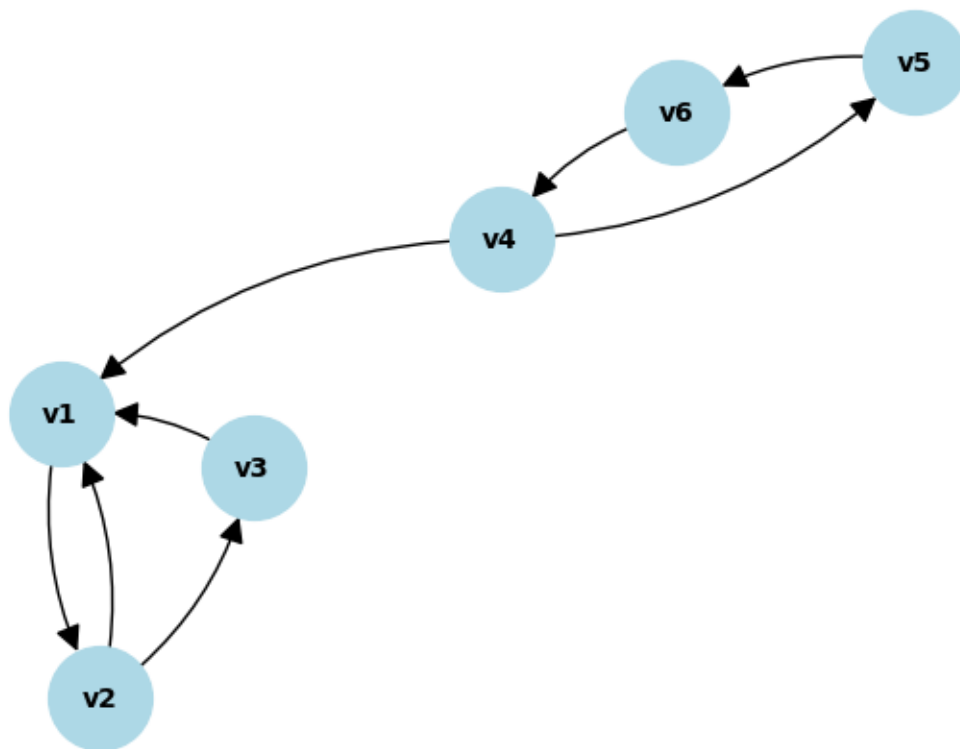


Figure 1: Example input graph.