# Advanced Coding Practice
# HW Problem #4

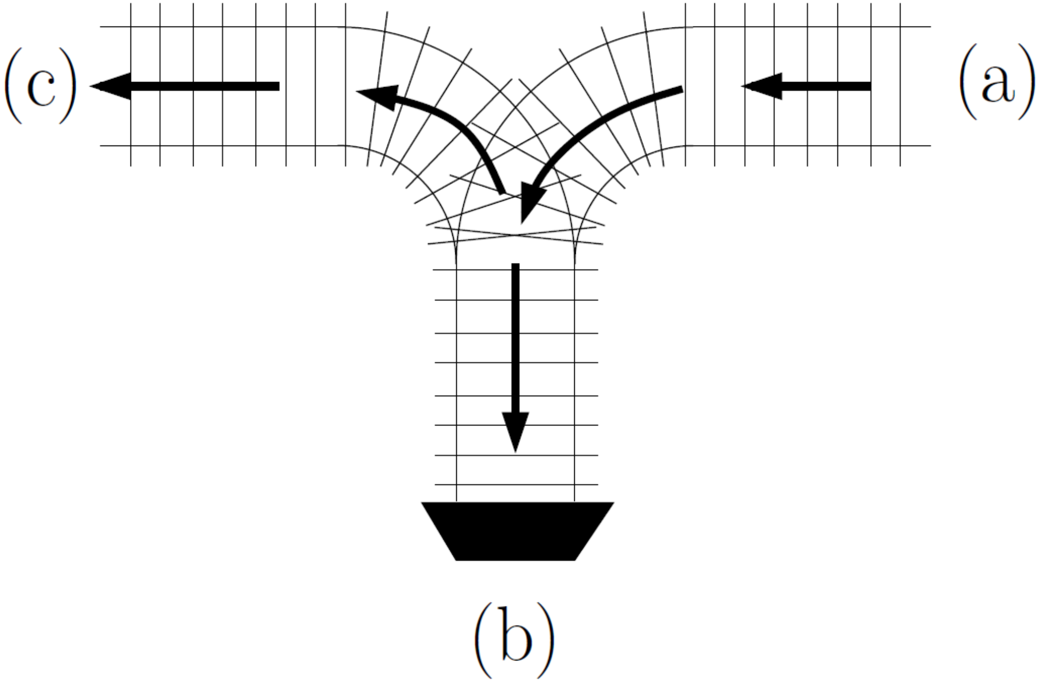
## 2023 Fall, CSE4152

Sogang University

# Train Rearrangement

There is a T-shaped railroad. Imagine that trains coming from (a), passing by (b), going to (c). We want to re-order the trains. For example, The order of coming trains is 3214. Rearranging the sequence of trains to 1234 is possible by following sequence of operations.

1. No. 3 train to (b).
2. No. 2 train to (b).
3. No. 1 train to (b).
4. No. 1 train to (c).
5. No. 2 train to (c).
6. No. 3 train to (c).
7. No. 4 train to (b).
8. No. 4 train to (c).



# Train Rearrangement (cont’d)

1. Is it possible to rearranging N trains coming from (a) in arbitrary order to increasing order like 123 · · ·N?

2. Please propose a method that figure out whether the given train order could be rearranged in increasing order after passing the T-railroad.

# Input & output

**Input**

3 // 3 following cases

5 1 2 3 4 5 // 5 trains and the train order.

4 3 2 1 4

5 3 2 5 4 1

**Output**

POSSIBLE

POSSIBLE

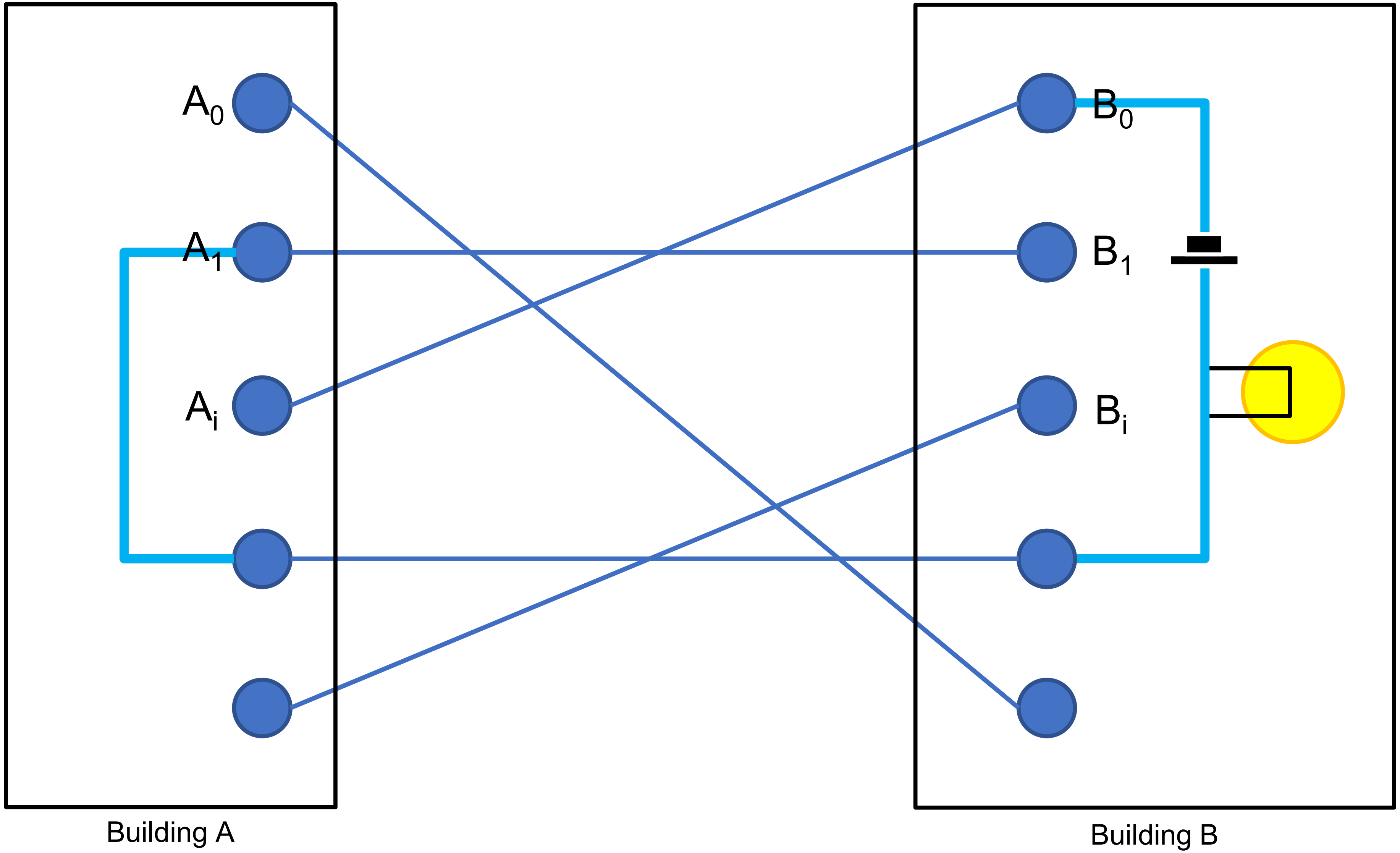
IMPOSSIBLE

# Agony of Engineer

Two extremely tall buildings A and B are very far apart from each other. There are N electrical lines connecting these buildings such that one terminal (end point) of each line is in building A while the other is in building B. All N terminals in each building are in a control box on the top of the building. You may assume that N is greater than two. Unfortunately, there are no elevators available in both buildings. An engineer wants to figure out how the terminals in one building are connected to the others by the electrical lines. He/she may use a light bulb, a battery, and electrical wires to check the connectivity among two or more lines. Suggest an algorithm for finding the connectivity between terminals while minimizing the number of travels between buildings. The engineer has to work alone in control boxes and is not allowed to install any additional electrical lines between buildings. Your answer should contain

- the minimum number of travel and
- Detailed description on how the engineer can do it.

# Agony of Engineer (cont'd)



# Agony of Engineer (cont’d)

We have three code files: **main.cpp, EnvGame.h, and solve.cpp**.

Class “EnvGame” is a simulator of the problem. You can use following methods of the EnvGame object to implement your proposed method.

**- connectWire(A, B):** connect a wire between two end nodes A and B on the current building (the building on which the engineer (you) stands

**- bool checkConnectivity(A, B):** connect wires, a lamp and a battery serially between end-node A and B (on the current building). If the lamp is on, return **true.** Otherwise, return **false.**

**- cleanAllWires():** clean all wires on the current building.

**- gotoOtherSide():** go to the opposite building.

# Agony of Engineer (cont’d)

Note that the engineer is at building A at the beginning. To check whether your solution is correct, please call function **answer()** in **EnvGame**.

**bool answer(vector<int> connectivityFromAtoB):** send the **connectivity from end-nodes of A to end-nodes of B** and check if the answer is true. If not, it returns false.

For the example figure, {4, 1, 0, 3, 2} is the right answer.

# Agony of Engineer (cont’d)

Your task is to implement function solve() in **solve.cpp** file.

You don’t need to modify any other files.

There is an example code that shows how to implement your solution.

Good Luck!