

# Algorithm

## Assignment 02

### Contest Hints

#### Cycle of Edges

**Explanation:**

আপনাকে একটি আনডাইরেক্টেড গ্রাফ দেওয়া থাকবে যেখানে  $N$  সংখ্যক নোড আর  $E$  সংখ্যক এজ দেওয়া থাকবে। আপনাকে বলতে হবে কয়টা এজ সাইকেল বানায়।

দ্রষ্টব্য: ডুপ্লিকেট এজ থাকবে না ইনপুট হিসেবে। নোড এর ভ্যালু ১ থেকে  $N$  এর মধ্যে হবে।

**Hints**

প্রথমে নোড সংখ্যা আর এজ সংখ্যা ইনপুট নিবেন। তারপর এজ অর্দি লুপ চালিয়ে দুইটা নোড ইনপুট নিবেন প্রতিবার। এরপর এই নোড দুইটির লিডার বের করবেন। যদি দুই নোড এর লিডার সেইম হয় তাহলে সেখানে একটি কাউন্টার ভেরিয়েবলের মান এক করে বাড়াবেন। আর সেইম না হলে নোড দুইটির ইউনিয়ন করে দিবেন। এরপর লুপের বাইরে এসে কাউন্ট এর ভ্যালু প্রিন্ট করে দিবেন।

# Building

## Explanation:

আপনি সবেমাত্র একটি ISP ব্যবসা খুলেছেন এবং আপনি আপনার এলাকায় আপনার ওয়াইফাই লাইন সংযোগ করতে চান। আপনার এলাকায় সেই বিল্ডিংগুলির সাথে সংযোগকারী রাস্তা রয়েছে। রাস্তাগুলো স্পষ্টতই দুই দিকে। প্রতিটি রাস্তায় তারের সংযোগ খরচ আছে। আপনি এমনভাবে সমস্ত বিল্ডিং সংযোগ করতে চান যাতে কোনও বিল্ডিং থেকে অন্যের সাথে সংযোগ থাকে, সরাসরি হওয়ার প্রয়োজন নেই।

যেহেতু আপনি একজন ব্যবসায়ী, আপনি চান মোট খরচ সর্বনিম্ন হোক। আপনি কি কাজটি করতে সর্বনিম্ন মোট খরচ বলতে পারেন

দ্রষ্টব্য: এখানে এক বিল্ডিং থেকে অন্য বিল্ডিং একাধিক রাস্তা হতে পারে। সমস্ত বিল্ডিং সংযোগ করা সম্ভব না হলে,  $-1$  প্রিন্ট করুন। প্রতিটি বিল্ডিং থেকে একটি সংখ্যা আছে  $1$  থেকে  $N$  এর মধ্যে।

## Hints

মিনিমাম স্প্যানিং ট্রি এর মাধ্যমে চিন্তা করব। প্রথমে এজ লিস্ট বানিয়ে সেটাকে সর্ট করে নিব। তারপর সেই লিস্ট থেকে এজ গুলোকে নিয়ে তাদের লিডার চেক করব। যদি সমান হয় তাহলে **continue** করে যাবো। আর নাহয় সেই এজ এর কস্টটা টোটালকস্ট নামের ভেরিয়েবলের সাথে যোগ করব এবং নোড দুইটির মাঝে ইউনিয়ন করে দিব সাথে একটি কাউন্টার ভ্যালু রাখবো যার মান এক করে বাড়াবো। শেষে চেক করব কাউন্টার এর ভ্যালু  $n-1$  এর সমান কিনা। যদি হয় তাহলে টোটাল কস্ট প্রিন্ট করে দিব আর নাহয়  $-1$  প্রিন্ট করব।

# Area of Component

## Explanation:

আপনাকে একটি 2D আকারের ম্যাট্রিক্স  $N \times M$  সাইজের দেওয়া হবে যেখানে শুধুমাত্র ডট(.) এবং বিয়োগ(-) থাকবে যেখানে ডট(.) মানে আপনি সেই ঘরে যেতে পারবেন এবং বিয়োগ(-) মানে আপনি পারবেন না।

আপনি মাত্র 4টি দিকে (উপর, নিচে, বাম এবং ডান) যেতে পারেন।

একটি কম্পোনেন্টের ক্ষেত্রফল হল সেই কম্পোনেন্টে থাকা ডট(.) সংখ্যা যা অ্যাক্সেসযোগ্য। আপনাকে সমস্ত উপলব্ধ উপাদানের ন্যূনতম এলাকা বলতে হবে।

দ্রষ্টব্য: কোন উপাদান না থাকলে, প্রিন্ট -1.

## Hints

প্রথমে সারি ও কলাম নাম্বার ইনপুট নিয়ে 2D গ্রিডটা ইনপুট নিব। তারপর গ্রিড এর প্রতি সেল এ যাবো এবং যদি সেলটি আনভিজিটেড হয় ও সেল এর ইন্ডেক্স - থাকে তাহলে Dfs কিংবা Bfs ফাংশনকে কল দিব। সেই ফাংশনে যখন এই ভ্যালিড চাইল্ড নোড এর কাছে যাবো ও কাউন্টার এর ভ্যালু ১ করে বাড়াবো। এরপর যেখানে সেই ফাংশন রিটার্ন করবে সেখানে একটা মিনিমাম বের করার ভ্যালু যেটা আগে INT\_MAX দিয়ে ইনিশিয়ালাইজ করা ছিলো সেটার মধ্যে মিনিমাম আপডেট করবো। তারপর সবশেষে সেই মিনিমাম ভ্যালু এর মান চেক করবো। যদি INT\_MAX থেকে যায় তাহলে -1 প্রিন্ট করবো আর নাহয় সেই ভ্যালু প্রিন্ট করবো।

# Emperor Alexander

### Explanation:

সম্রাট আলেকজান্ডার তার বিশাল সাম্রাজ্যের শহরগুলির সাথে সংযোগকারী  $N$  রাস্তাগুলির একটি বিস্তৃত নেটওয়ার্কের কল্পনা করছেন। এটি অর্জনের জন্য, তিনি প্রতিটি নির্মাণ ব্যয় সহ  $E$  সংখ্যক রাস্তা নির্মাণের পরিকল্পনা তৈরি করেছেন। ঘনিষ্ঠভাবে পরীক্ষা করার পরে, এটি তার নজরে এসেছে যে এই রাস্তাগুলির মধ্যে কিছু অপ্রয়োজনীয় খরচ হতে পারে। তিনি এই ব্যয়যোগ্য রাস্তাগুলি সনাক্ত করতে সর্বনিম্ন মোট নির্মাণ ব্যয় নির্ধারণে আপনার হেল্প চেয়েছেন। যদি এই পরিকল্পনাটি উপলব্ধি করা অসম্ভব হয় তবে **"Not Possible"** প্রিন্ট করবেন।

আপনি কি সম্রাট আলেকজান্ডারকে খরচ অপ্টিমাইজ করার পরিকল্পনা থেকে সরানো যেতে পারে এমন রাস্তার সংখ্যা খুঁজে বের করতে এবং ন্যূনতম মোট নির্মাণ ব্যয় গণনা করতে সহায়তা করতে পারেন?

দ্রষ্টব্য: সেই পরিকল্পনায় দুটি শহরের মধ্যে একাধিক রাস্তা থাকতে পারে। কিন্তু শহর  $A$  থেকে  $A$  পর্যন্ত কোন রাস্তা থাকবে না।

### Hints

এই প্রব্লেমটি আমরা **MST** দিয়ে করবো। প্রথমে **number of nodes and number of edges** ইনপুট নিব। এরপর **edge** গুলোকে ইনপুট নিয়ে **edgelist** বানাবো। এরপর **MST** বানাবো এবং সেইটির মাধ্যমে **totalcost** বের করবো, এর সাথে **MST** তে **edge count** করবেন। এরপর **input** নেয়া **number of edge** থেকে **MST** তে **count** করা **edge** বাদ দিলে সেইটি হবে **number of roads need to**, এর সাথে **totalcost print** করবো।

## Maze

### Explanation:

রেজিয়া  $R$  অবস্থান থেকে শুরু করে  $N \times M$  আকারের একটি **2D maze** এ আটকা পড়েছে, এবং তার লক্ষ্য হল  $D$  দ্বারা চিহ্নিত প্রস্থানে পৌঁছানো। **Maze** এ **#** দ্বারা রিপ্রেজেন্ট করা ব্লক রয়েছে এবং শুধুমাত্র বিন্দু (.) দ্বারা চিহ্নিত সেল এর মধ্য দিয়ে যেতে পারে সে। তাকে যত তাড়াতাড়ি সম্ভব পালাতে হবে, সে যে পথ অনুসরণ করবে তা আমাদের নির্ধারণ করতে হবে।

Maze থেকে প্রশ্নান করার জন্য রেজিয়ার রুট প্রতিনিধিত্বকারী প্রতিটি কক্ষে একটি X রাখুন। যদি তার প্রশ্নান করার জন্য কোনও কার্যকর পথ না থাকে তবে Maze টি অপরিবর্তিত রেখে দিন।

দ্রষ্টব্য: রেজিয়া চারটি দিকে যেতে পারে - ডান, বাম, উপরে এবং নীচে। নির্দিষ্ট ক্রম মেনে চলা অত্যন্ত গুরুত্বপূর্ণ: প্রথমে ডানে, তারপরে বামে, তারপরে উপরে এবং শেষ পর্যন্ত নিচের চেষ্টা করতে হবে অবশ্যই।

### Hints

প্রথমে **row and column** কে ইনপুট নিব। এরপর **2D grid** টি ইনপুট নিব, এরপর **source and destination** এর ইনডেক্স বের করবো। **Source** কে দিয়ে **bfs** কল করবো। **bfs** এর মধ্যে **parent** কে বের করে রাখবেন। **bfs function complete** হওয়ার পর **parent** এর পর মাধ্যমে **destination** কে ব্যবহার করে **shortest path** এর ইনডেক্সে 'X' assign করে দিব। এরপর **2D grid** টিকে প্রিন্ট করে দিব।