Mentor - Kumar K(https://www.linkedin.com/in/kumark1/)

Join the premium daily live high quality training -> https://training.designa.in By Kumar K

Atlassian | Software Engineer | CTC 70+ LPA

Anonymous User

6693 Oct 02, 2024 Oct 02, 2024

Atlassian Interview

DSA Round

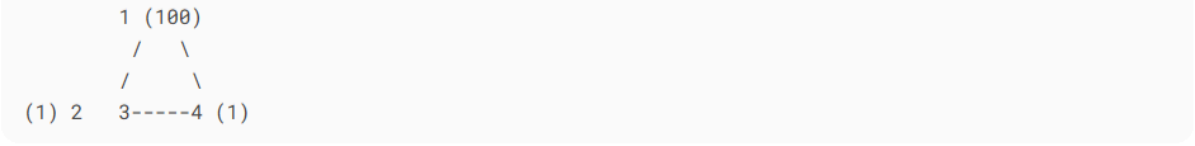
Problem Statement : You're designing a resilient communication network between N distributed nodes (islands), each with a unique importance score. Initially, every node is fully connected to every other node. Due to cost constraints (C), Atlassian's infrastructure optimization system can decommission certain high-cost links. Your task is to compute the minimum number of nodes accessible from the primary node (Node 1) after strategic disconnections.

Graph Before Destruction:

Copy Download

Graph After Destruction:

Copy Download



(Edges `1-3`, `1-4`, and `3-4` remain.)

-> **Interview Problem: Optimally Disconnecting Atlassian Islands**

Context: You're working on a simulation tool at Atlassian that manages a network of islands connected by bridges. Initially, every island is connected to every other island. Each island has an "importance value" associated with it.

A malicious agent, **Alex** (an Atlassian engineer), can destroy some bridges between islands, but only if the total destruction cost remains within a given budget. The cost to destroy the bridge between island i and island j is `importance[i] * importance[j]`.

Your colleague **Jordan** (an Atlassian employee) lives on island 1 and enjoys visiting other islands. After **Alex** destroys some bridges optimally to disconnect the network, we want to determine how many islands **Jordan** can still access, including his own.

🧩 **Problem Statement:** You are given:

- An integer `N` — the number of islands.
- An array `A` of length `N`, where `A[i]` is the importance of the i-th island (1-indexed).
- A long integer `C` — the maximum total cost **Alex** can spend to destroy bridges.

Each island is initially connected to every other island. Your task is to compute the minimum number of islands (including island 1) that **Jordan** can access after **Alex** destroys bridges optimally (to minimize the size of the connected component that contains island 1).

🗒️ **Input:**

- First line: integer `T` — number of test cases.

For each test case:

- Line 1: Two integers `N` and `C`.
- Line 2: `N` integers `A[1] A[2] ... A[N]` — importance values.

📤 **Output:** For each test case, output the minimum number of islands (including island 1) **Jordan** can visit

Analysis :-> What can you do to make sure only "N-1" nodes are visited in total if you start at node-1

-> Choose node “i” and break its connection with all other nodes so only “N-1” nodes stay connected to each other in the component with node - “1”

-> Hence if you start travelling from node “1” you can visit all N-1 nodes except node “i”

-> possible answer => a[2]*(sum of all other numbers) -> when node-’2’ is removed.

OR

=> a[3]*(sum of all other numbers) -> when node-’3’ is removed.

OR .

.

.

.

.

.

Min of all possibilities <= C -> then yes it is possible to get answer as N-1

-> Now try for N-2;

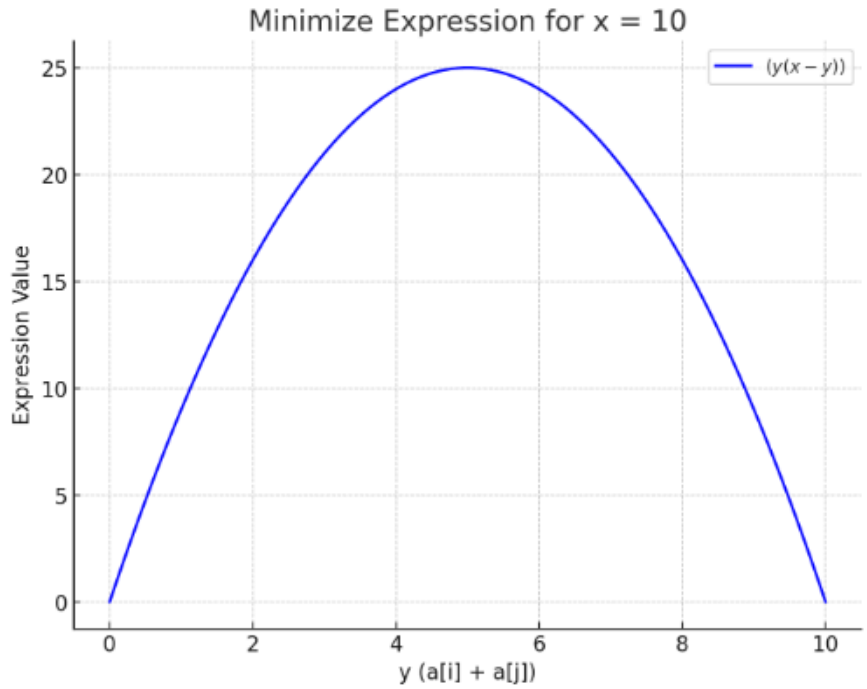
-> (i,j) will be selected from [2….N]

-> minimize (a[i]+a[j])*(sum of rest of the numbers)

-> x = sum of all numbers; y = a[i]+a[j];

-> minimize (y*(x-y))

Graph created.



-> Hence "y" can only be either sum of two largest numbers in range [2..N] or two smallest numbers in range [2….N]

-> do it and minimize;

-> Same technique if you try to get answer as N-3; N-4; ….. 1

Implementation :->

C++ https://ideone.com/hZDGhb

Java.

Py