TIME REMAINING
0:36:04

End Exam

Note: Please **SUBMIT** each question individually before ending the exam to receive score.
Note: This is a monitored test.

Duplicate Product IDs E-Commerce - Git

Background: You are a software engineer at ShopSmart, a rapidly growing e-commerce platform that relies on a system of unique Product IDs to manage its vast inventory. However, due to overlapping product catalogs from different suppliers, some product IDs have become duplicated, which results in issues when updating or managing the database. Currently, the system rejects updates if the Product IDs are not unique. Your task is to ensure all Product IDs are unique. Updating a Product ID incurs a cost, which is equal to the absolute difference between the new Product ID and the old Product ID. For example

Old Product ID = 300

New Product ID = 230

Cost = |230 - 300| = 70

Additionally, since ShopSmart uses Git for version control, you must resolve merge conflicts between two branches (master and stage) in a structured manner before processing the data.

Task:

- Clone the Git Repository (https://gitlab-fg2025.arbisoft.com/freshgrad2025/HireSprint2025)
    - Clone the repository and switch to the main branch.

- Merge the stage Branch into main
    - The repository contains two branches: main and stage. Each branch has two files: inventory1.txt and inventory2.txt. When merging stage into main, conflicts will arise in both files.
- Resolve Merge Conflicts Using the Following Rules
    - For inventory1.txt
        - Only keep products that are common in both branches.
        - The first common product ID should be picked from main, the second from stage, the third from main, and so on.
    - For inventory2.txt
        - Only keep products that are common in both branches.
        - The first common product ID should be picked from stage, the second from main, the third from stage, and so on.
    - The order of the products should remain the same.
    - Create a final file, “inventory.txt”, with the following content
        - First inventory1.txt content (without first line total number of test case integer)
        - Then append inventory2.txt (without first line total number of test case integer)
        - Add/update first line as total number of testcases
- Submit Proof of Git Commands Used
    - After resolving the conflicts and running the script, take a screenshot of your terminal history using:

- Linux/macOS:
  - cat ~/.bash_history | tail -n 50 (bash)
  - cat ~/.zsh_history | tail -n 50 (zsh)
- Windows (PowerShell): Get-History
- Windows (Git Bash): history

- Submit the screenshot as proof of using Git properly.

- Write a function that calculates the minimum cost required to make all Product IDs unique.
  - Your solution should handle large amounts of data efficiently, as duplicate IDs may arise frequently due to new products being uploaded from various suppliers. Since this problem needs to be resolved as part of the regular product ingestion pipeline, the solution must be scalable and quick.

Problem Constraints:

- Product IDs are non-negative integers.
- The list of Product IDs can be very large (up to millions of products), so an efficient algorithm is required.

Approach:

You need to implement an algorithm that processes a list of Product IDs and ensures each ID is unique with a minimal total cost. The function should output the minimal total cost incurred for resolving the conflicts.

Format of input (after resolving conflicts):

Number of products

ProductName-ProductId

ProductName-ProductId

ProductName-ProductId

Format of expected output:

Minimum total cost integer

Git Example

inventory1.txt in master:

4

ProductA-1

ProductB-3

ProductC-5

ProductD-7

inventory1.txt in stage:

5

ProductA-2

ProductB-3

ProductC-4

ProductD-5

ProductE-8

After Resolving the Conflict (Final inventory1.txt in master):

4

ProductA-1 # From main

ProductB-3 # From stage

ProductC-5 # From main

ProductD-5 # From stage

inventory2.txt in master:

4

ProductX-10

ProductY-15

ProductZ-20

ProductW-25

inventory2.txt in stage:

5

ProductX-10

ProductY-12

ProductZ-15

ProductW-22

ProductV-30

After Resolving the Conflict (Final inventory2.txt in master):

4

ProductX-10 # From stage

ProductY-15 # From main

ProductZ-15 # From stage

ProductW-25 # From main

Final Input File for Calculation

8

ProductA-1

ProductB-3

ProductC-5

ProductD-5

ProductX-10

ProductY-15

ProductZ-15

ProductW-25

Your Response LOADING