

# Merging Related Legal Documents

## 1 Problem Statement

You are building a document management system for an Indian law firm. Legal documents (petitions, applications, affidavits, etc.) often reference each other or belong to the same legal matter. For example:

- Document 5 might reference Document 2
- Document 2 might reference Document 0
- This means Documents 0, 2, and 5 all belong to the same legal matter/thread

Given pairs of documents that belong to the same legal matter, assign each document a canonical matter ID. Documents in the same matter should share the same matter ID, which should be the smallest document index in that matter.

**Context:** In Indian legal practice, lawyers need to group related documents together. For example, a writ petition, its supporting affidavits, counter-affidavits, and court orders all belong to the same matter. The canonical ID helps identify the primary document (usually the first one filed) in each matter.

### 1.1 Input Format

```
{  
  "n": 5,  
  "pairs": [  
    {"document1": 0, "document2": 1},  
    {"document1": 2, "document2": 3},  
    {"document1": 1, "document2": 2}  
  ]  
}
```

- **n**: Total number of documents (indexed 0 to n-1)
- **pairs**: Array of pairs indicating two documents belong to the same legal matter
- Each pair: {"document1": i, "document2": j} means documents i and j are related

### 1.2 Output Format

```
[0, 0, 0, 0, 4]
```

Return an array of length **n** where **result[i]** is the canonical matter ID for document **i**. The canonical ID is the smallest document index in that matter's group.

## 2 Example Test Cases

### 2.1 Test Case 1: Simple Document Grouping

- **Input:**
  - n = 3
  - pairs = ["document1": 0, "document2": 1]
- **Output:** [0, 0, 2]
- **Explanation:** Documents 0 and 1 belong to the same matter (canonical ID: 0, since 0 ∪ 1). Document 2 is separate (canonical ID: 2)

### 2.2 Test Case 2: Chain of Related Documents

- **Input:**
  - n = 4
  - pairs = ["document1": 0, "document2": 1, {"document1": 1, "document2": 2}]
- **Output:** [0, 0, 0, 3]
- **Explanation:** Documents 0-1 are related, and 1-2 are related, so 0-1-2 form one matter (canonical ID: 0). Document 3 is separate (canonical ID: 3)

### 2.3 Test Case 3: Multiple Legal Matters

- **Input:**
  - n = 5
  - pairs = ["document1": 0, "document2": 1, {"document1": 2, "document2": 3}]
- **Output:** [0, 0, 2, 2, 4]
- **Explanation:** Two separate matters—Documents 0-1 form Matter 0, Documents 2-3 form Matter 2, Document 4 is standalone (Matter 4)

### 2.4 Test Case 4: All Documents Connected

- **Input:**
  - n = 4
  - pairs = ["document1": 0, "document2": 1, {"document1": 2, "document2": 3}, {"document1": 1, "document2": 2}]
- **Output:** [0, 0, 0, 0]
- **Explanation:** All documents are connected (0-1, 2-3, and 1-2 link them), so they form one matter with canonical ID 0

## 2.5 Test Case 5: No Relationships

- **Input:**
  - $n = 3$
  - pairs = []
- **Output:** [0, 1, 2]
- **Explanation:** No pairs provided, so each document gets its own matter ID

## 3 Edge Cases

- No pairs provided → each document gets its own matter ID (equal to its index)
- Single document ( $n = 1$ ) → [0]
- All documents in one matter → all get the smallest index (0)
- Repeated pairs (same pair multiple times) → should handle gracefully
- Transitive relationships → if A-B and B-C, then A, B, C should share same matter ID

## 4 Requirements

- Documents in the same matter must share the same canonical matter ID
- The canonical ID must be the smallest document index in that matter
- For optimal performance, use an efficient data structure that supports union and find operations
- Stable output: same input should always produce same output
- Handle large inputs efficiently (near-linear time complexity)