

Not the Best: একটি shortest path problem

সমস্যা বর্ণনা (Problem Description)

এই সমস্যা হলো weighted shortest path বা minimum cost path খুঁজে বের করার সমস্যা।

তোমাকে দেওয়া থাকবে কিছু node এবং weighted edges। লক্ষ্য:

Source node থেকে destination node পর্যন্ত সর্বনিম্ন মোট ওজন (minimum total weight) দিয়ে পৌঁছানো।

বিশেষ দিক:

- প্রতিটি edge-এর weight ধনাত্মক (positive)
- Graph-এ cycles থাকতে পারে
- “Not the Best” সমস্যা বলে বোঝায় shortest path খুঁজতে হবে কিন্তু সব node traverse করা লাগবে না

গ্রাফের বিবরণ (Graph Details)

- N node, M edge
- $\text{edge}[i] = (u, v, w)$, যেখানে w = edge weight
- directed বা undirected উভয় হতে পারে
- লক্ষ্য: source \rightarrow destination node পর্যন্ত shortest path

কেন Dijkstra? (Why Dijkstra?)

- গ্রাফের edge weight ধনাত্মক
- BFS ব্যবহার করলে shortest path ওজনের ভিত্তিতে সঠিক হবে না
- তাই Dijkstra অ্যালগরিদম ব্যবহার করা হয়
- Priority queue ব্যবহার করে সর্বদা current shortest distance node প্রসেস করা হয়

সমাধানের বৈশিষ্ট্য (Features)

- Weighted graph-এর জন্য উপযুক্ত
- Efficient due to priority queue
- Directed বা undirected উভয় graph এ ব্যবহারযোগ্য
- Positive edge weight নিশ্চিত করলে সবসময় সঠিক ফলাফল

Pseudocode

```
FUNCTION dijkstra(graph, start, n):  
    DEFINE dist[1..n] = INFINITY  
    DEFINE visited[1..n] = False  
    DEFINE priority_queue pq  
  
    dist[start] = 0  
    PUSH (0, start) INTO pq  
  
    WHILE pq IS NOT EMPTY:  
        (d, u) = POP pq  
        IF visited[u]:  
            CONTINUE  
        visited[u] = True  
  
        FOR each (v, w) in graph[u]:  
            IF dist[u] + w < dist[v]:  
                dist[v] = dist[u] + w  
                PUSH (dist[v], v) INTO pq  
  
    RETURN dist
```