



## School Road (Solution)

Review tatyam

Proof noshi91

In this task, given an undirected weighted graph with  $N$  vertices and  $M$  edges, we need to determine whether there exist more than one values of the lengths of simple paths (paths without visiting the same vertex more than once) from  $S = (\text{vertex } 1)$  to  $T = (\text{vertex } N)$ .

### Subtask 1 ( $M \leq 40$ )

We first consider a solution to search all the possibilities.

Let us calculate all the simple paths starting from  $S$  by DFS. What is the time complexity?

Let  $P(M)$  be the maximum of “the number of simple paths starting from  $S$ ” for an undirected graph with  $M$  edges. By a case-by-case analysis according to the degree of the vertex  $S$ , we see that

$$P(M) \leq 1 + \max_{1 \leq d \leq M} \{d \times P(M - d)\}$$

holds. We estimate the upper bound using this recurrence formula. Then we have  $P(40) \leq 4.2 \times 10^6$ . Therefore, this subtask can be solved by DFS whose time complexity is  $O(M \cdot P(M))$ .

### Subtask 2 ( $N \leq 18$ )

If there exist multiple edges, we can use only one of them to make a simple path. We shall replace multiple edges by an edge.

Since we are finding paths with different lengths, we do not need several edges of the same length. We only need two edges of different lengths. Therefore, for each pair of vertices, we only need an edge of minimum length and an edge of maximum length. Then, the total number of edges becomes  $O(N^2)$ .

After that, it is enough to calculate a shortest simple path and a longest simple path. We can calculate a shortest simple path by Dijkstra’s algorithm whose time complexity is  $O(N^2)$ . But we cannot calculate a longest simple path by the same method. Since  $N \leq 18$ , we can calculate it by bitDP. We put

$$\text{dp}[bit][x] = \left( \begin{array}{l} \text{the maximum and the minimum lengths of simple paths passing through} \\ \text{the set of vertices } bit \text{ from } S \text{ to the vertex } x \end{array} \right).$$

We can solve this subtask in  $O(2^N N^2 + M)$  time.



### Subtask 3 ( $M - N \leq 13$ )

In this subtask,  $M$  is close to  $N$ . This means the average degree of the vertices is close to 2, and more than half of the vertices have degree  $\leq 2$ .

Except for  $S$  and  $T$ , we will never use a vertex of degree 1. Thus, we can delete it and the edge connecting with it. Repeating this, we may assume that almost all vertices have degree 2.

Except for  $S$  and  $T$ , if one visits a vertex of degree 2 by passing through an edge, one will leave it by passing through the other edge. Thus, we can regard a vertex of degree 2 together with the two edges as an edge. We replace a vertex of degree 2 together with the two edges by an edge. Repeating this, we may assume that all the vertices other than  $S, T$  have degree  $\geq 3$ .

By the above processes, the graph becomes a graph satisfying  $N \leq 30$ ,  $M \leq 43$ ,  $M - N \leq 13$ . Since  $M \leq 43$  is close to the constraints of Subtask 1, we can solve it using the solution of Subtask 1.

When  $M \leq 43$ , one may wonder if “the number of simple paths (of length  $\geq 0$ ) starting from  $S$ ” becomes larger than Subtask 1. However, due to the constraints  $M - N \leq 13$ , this value actually cannot be larger than Subtask 1.

### Subtask 4 (if $a \neq b \neq c$ , then there exists an $a - c$ path without passing through the vertex $b$ )

What is the meaning of this constraint?

If there exists a triple  $a, b, c$  such that every  $a - c$  path passes through the vertex  $b$ , then  $a - c$  becomes disconnected if we delete the vertex  $b$ . In other words, the vertex  $b$  is an articulation point. Conversely, if there exists an articulation point, there exists a triple  $a, b, c$  such that every  $a - c$  path passes through the vertex  $b$ . Therefore, the condition that such a triple does not exist means every vertex is not an articulation point. Hence the graph is 2-vertex-connected.

Assume that the graph is 2-vertex-connected. Then for every edge  $i - j$ , there exists a simple  $S - T$  path containing the edge  $i - j$ .

**Proof** We add the vertices  $a, b$  and the edges  $S - a, T - a, i - b, j - b$ . After that, the graph is still 2-vertex-connected. There exist two simple paths from  $a$  to  $b$  which do not share a common vertex except for the end points (Menger’s theorem). We delete the vertices  $a, b$  from the two simple paths and add the edge  $i - j$  to connect them. Then we obtain a desired simple path.

Therefore, if there exist multiple edges with different lengths, the answer is 1. If there exist multiple edges



with the same length, we may replace them by an edge. We may assume that the graph is simple.

By the same way as in Subtask 3, we compress the vertices of degree 2. We may assume that all the vertices other than  $S, T$  have degree  $\geq 3$ .

After that, if only the edge  $S - T$  remains, the answer is obviously 0. Otherwise, one may wonder if the answer is always 1. We shall prove it.

### Prerequisite conditions

- The graph is simple and 2-vertex-connected.
- All the vertices other than  $S, T$  have degree  $\geq 3$ .
- There exist at least 4 vertices. (The above condition cannot be satisfied if there exist less than 4 vertices.)

Let  $\text{dist}(i, j)$  be the minimum distance between  $i$  and  $j$ . We may assume that every edge  $i$  satisfies

$$\text{dist}(S, A_i) + C_i + \text{dist}(B_i, T) = \text{dist}(S, T)$$

or

$$\text{dist}(S, B_i) + C_i + \text{dist}(A_i, T) = \text{dist}(S, T).$$

Otherwise, there exists an edge which does not satisfy this condition. Then, there exists a simple  $S - T$  path containing it, and the answer is 1.

We give orientations to the edges so that every edge moves farther from the vertex  $S$  (i.e., every edge gets close to the vertex  $T$ ). If the indegree of a vertex is larger than the outdegree, we mark it as red. Other vertices are colored blue.

Let  $P$  be a path from  $S$  to  $T$  passing through **the maximum number of edges**. Since  $P$  contains at least 4 vertices and it passes through the maximum number of edges, the vertex on the path  $P$  next to  $S$  is blue and the vertex just before  $T$  is red. There exists an edge on the path  $P$  which changes the color from blue to red. Let  $u \rightarrow v$  be such an edge.

Here the vertex  $u$  is blue, and the vertex  $v$  is red. There exist edges  $u \rightarrow x, y \rightarrow v$  other than the edge  $u \rightarrow v$ . We take shortest paths  $S \rightarrow y$  and  $x \rightarrow T$ . We consider the path  $S - y - v - u - x - T$ . The path  $S - y - v - u - x - T$  is simple (if it is not simple, then there exists a path whose number of edges is larger than  $P$ ), and its length is larger than  $\text{dist}(S, T)$ .

Therefore, the answer is 1.

In summary, after replacing multiple edges by an edge and contracting vertices of degree 2, if only the edge  $S - T$  remains, the answer is 0. If there exist multiple edges with different lengths or an edge other than  $S - T$  remains, the answer is 1. The time complexity is  $O(M)$ .



## Subtask 5

In this subtask, the difference from Subtask 4 is that there may exist articulation points. Also, there may exist a vertex such that if one visits it from  $S$ , then one cannot visit  $T$  afterward. We need to delete unnecessary parts from the graph.

We decompose the graph into 2-vertex-connected components. We only need 2-vertex-connected components between  $S$  and  $T$  because if one tries to pass through an edge outside such components, one has to pass through an articulation point more than once.

We add an edge of length  $\text{dist}(S, T)$  between  $S$  and  $T$ . Then, we only need the 2-vertex-connected component containing the  $S - T$  edge.

After deleting unnecessary parts from the graph, we can solve this subtask by the same way as Subtask 4. The time complexity is  $O(M)$ .