S242 - Path of Infinity

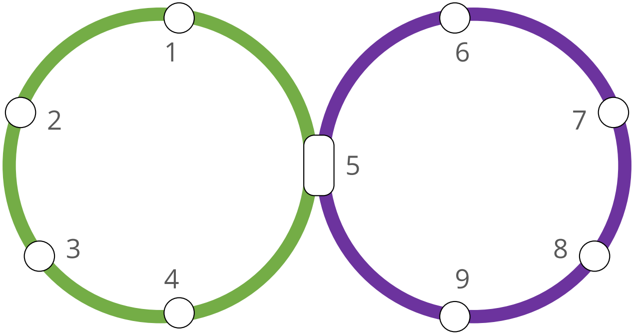
Kelvin Chow {Lrt1088}

2024-02-17

Table of Contents

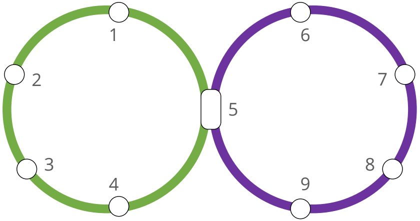
Background

Problem Idea by `mtyeung1`
Preparation by `Lrt1088`



Problem Restatement

- Given a graph with the structure shown in the figure, each ring has N nodes.
- Each node is assigned a weight C_i.
- Answer Q all-pairs shortest path queries. If the answer is negative, output 0.



Statistics

0 points	9	+	2	+	0	+	0	=	11
13 points	8	+	0	+	0	+	0	=	8
27 points	9	+	1	+	0	+	0	=	10
38 points	1	+	0	+	0	+	0	=	1
54 points	1	+	0	+	0	+	0	=	1
56 points	6	+	12	+	5	+	0	=	23
100 points	0	+	4	+	9	+	8	=	21

First solved by **ryanjz2024** at **21m 6s**

Subtasks

Subtask	Points	Constraints
1	13	$Q = 1, C_i \geq 1, 1 \leq X_1, Y_1 \leq N$
2	14	$Q = 1, C_i \geq 1$
3	29	$C_i \geq 1$
4	11	$Q = 1, C_i \geq -1, 1 \leq X_1, Y_1 \leq N$
5	16	$Q = 1, 1 \leq X_1, Y_1 \leq N$
6	17	No additional constraints

Table of Contents

Subtask 1 (13 pts): $Q = 1, C_i \geq 1, 1 \leq X_1, Y_1 \leq N$

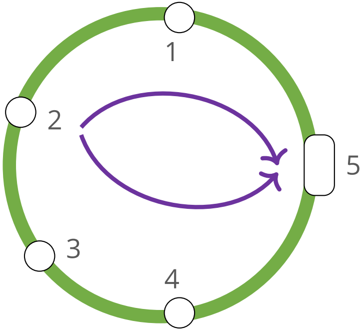
- Let's deconstruct the constraints.
  - $Q = 1$ allows us to write an $O(NQ)$ algorithm.
  - $C_i \geq 1$ eliminates the possibility of negative fares. (We always want to visit as few stations as possible!)
  - $1 \leq X_1, Y_1 \leq N$ means that we can focus on the ring on the left.

Subtask 1 (13 pts): $Q = 1, C_i \geq 1, 1 \leq X_1, Y_1 \leq N$

Observation: There are 2 paths from station X_1 to station Y_1.

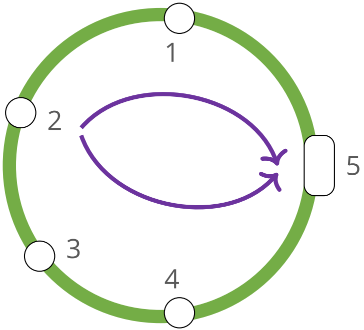
Example: Station 2 to Station 5.

- Path 1: $2 \to 3 \to 4 \to 5$ (Cost: $C_2 + C_3 + C_4 + C_5$)
- Path 2: $2 \to 1 \to 5$ (Cost: $C_2 + C_1 + C_5$)



Subtask 1 (13 pts): $Q = 1, C_i \geq 1, 1 \leq X_1, Y_1 \leq N$

- Assume $X_1 \leq Y_1$. The costs of the 2 paths are:
  - $C_{X_1} + C_{X_1+1} + \ldots + C_{Y_1}$
  - $(C_1 + C_2 + \ldots + C_N) - (C_{X_1+1} + C_{X_1+2} + \ldots + C_{Y_1-1})$



Subtask 1 (13 pts): $Q = 1, C_i \geq 1, 1 \leq X_1, Y_1 \leq N$

- Assume $X_1 \leq Y_1$. The costs of the 2 paths are:
  - $C_{X_1} + C_{X_1+1} + \ldots + C_{Y_1}$
  - $(C_1 + C_2 + \ldots + C_N) - (C_{X_1+1} + C_{X_1+2} + \ldots + C_{Y_1-1})$

Implementation:

```
cin >> X[1] >> Y[1];
if(X[1] > Y[1]) swap(X[1], Y[1]);
int route1 = 0;
for(int i=X[1]; i<=Y[1]; ++i) route1 += C[i];
int route2 = 0;
for(int i=1; i<=N; i++) route2 += C[i];
for(int i=X[1]+1; i<Y[1]; i++) route2 -= C[i];
cout << min(route1, route2) << "\n";
```

Subtask 2 (14 pts): $Q = 1$, $C_i \geq 1$

- In this subtask, we have to handle cases where X_i and Y_i belong to different railway lines.
- Decompose this into two different journeys, $X_i \to N$ and $N \to Y_i$. Each of them contains two different possible paths.

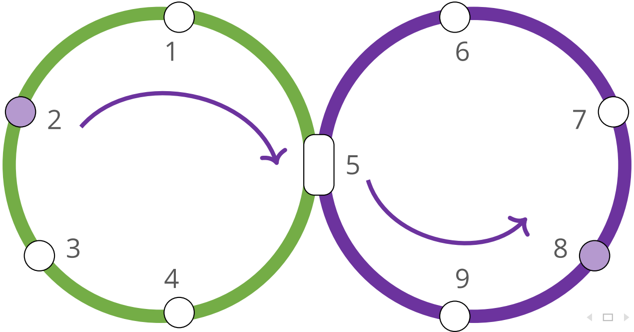


Table of Contents

Subtask 3 (29 pts): $C_i \geq 1$

- We're using a $O(NQ)$ solution in the previous subtasks, which will result in a TLE verdict.
- Luckily, the only slow part is for calculating the range sums. This can be optimized by **prefix sum / partial sum**.
- The time complexity becomes $O(N + Q)$ in total.

Table of Contents

Subtask 4 (11 pts): $Q = 1, C_i \geq -1, 1 \leq X_1, Y_1 \leq N$

- You will realize that the solution to subtasks 1-3 (assuming no **repeated vertices** are visited) does not work anymore. This is because we might obtain a (more) negative cost by going farther.
- In this subtask, we are only required to deal with $C_i = -1$.

Subtask 4 (11 pts): $Q = 1, C_i \geq -1, 1 \leq X_1, Y_1 \leq N$

Observation 4.1

If there are two adjacent vertices with $C_i = -1$, the answer is always 0.

Reason: We can reach the two vertices, visit them repetitively to obtain $-\infty$ cost.

Subtask 4 (11 pts): $Q = 1, C_i \geq -1, 1 \leq X_1, Y_1 \leq N$

Observation 4.2

If there are two adjacent vertices with $C_i = -1$ and $C_i = 0$ respectively, the answer is always 0.

Reason: We can reach the two vertices, visit them repetitively to obtain $-\infty$ cost.

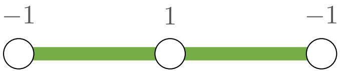
Subtask 4 (11 pts): $Q = 1, C_i \geq -1, 1 \leq X_1, Y_1 \leq N$

Observation 4.3

In all other cases, it's not wise to visit any vertex more than once.
Reason: We will be charged extra non-negative cost if we visit more vertices than what's required.

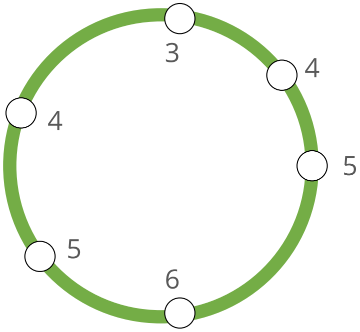
Subtask 4 (11 pts): $Q = 1, C_i \geq -1, 1 \leq X_1, Y_1 \leq N$

- By the above observations, we are only required to check for adjacent pairs of vertices. The answer is $-\infty$ if there are $\{-1, -1\}$ or $\{0, -1\}$ pairs. Remember to check **ALL** adjacent pairs especially for the pairs around N.
- Otherwise, apply subtask 2's solution to find the minimal cost.
- Be aware of the following case! The cost is -1 which is in between $-\infty$ and 0.



Subtask 5 (16 pts): $Q = 1, 1 \leq X_1, Y_1 \leq N$

- Subtask 4 should have given you a quick insight on the $-\infty$ case - the answer is $-\infty$ if and only if there exists a **negative cycle** in the graph.



Subtask 5 (16 pts): $Q = 1, 1 \leq X_1, Y_1 \leq N$

- Suppose the cycle has **even length**: $P_1 \to P_2 \to P_3 \to \cdots \to P_{2K} \to P_1$.
- Since the cycle is a negative cycle, we have $C_{P_1} + C_{P_2} + \cdots + C_{P_{2K}} < 0$.
- Group the vertices in the cycle into groups of 2. We have $(C_{P_1} + C_{P_2}) + \cdots + (C_{P_{2K-1}} + C_{P_{2K}}) < 0$.
- Since the sum of the K groups are negative, at least one group is negative, i.e. $C_u + C_v < 0$ for some adjacent (u, v).
- We can form a **length-2 negative cycle** by repeatedly visiting vertices u and v only.

Subtask 5 (16 pts): $Q = 1, 1 \leq X_1, Y_1 \leq N$

- What if the cycle is of **odd length**? We cannot group them anymore...
  Example: $1 \to 2 \to 3 \to 4 \to 5 \to 1$.
- We can produce an even length cycle from an odd length cycle by **repeating the list of vertices twice**.
  Example: $1 \to 2 \to 3 \to 4 \to 5 \to 1 \to 2 \to 3 \to 4 \to 5 \to 1$.
- The argument on the previous slide will still apply.

Subtask 5 (16 pts): $Q = 1, 1 \leq X_1, Y_1 \leq N$

Observation 5.1 (Summary)

If the graph contains a negative cycle, it also contains a negative cycle of **length 2**.

$$\Downarrow$$

Observation 5.2 (Corollary)

A graph contains a negative cycle if and only if there exists a negative cycle of length 2, i.e. there exists adjacent vertices (u, v) such that $C_u + C_v < 0$.

Full Solution

We've already covered all necessary details for the full solution!
You can get full score by combining the solutions to Subtask 3 (prefix sum / partial sum) and Subtask 5 (checking for negative cycles).