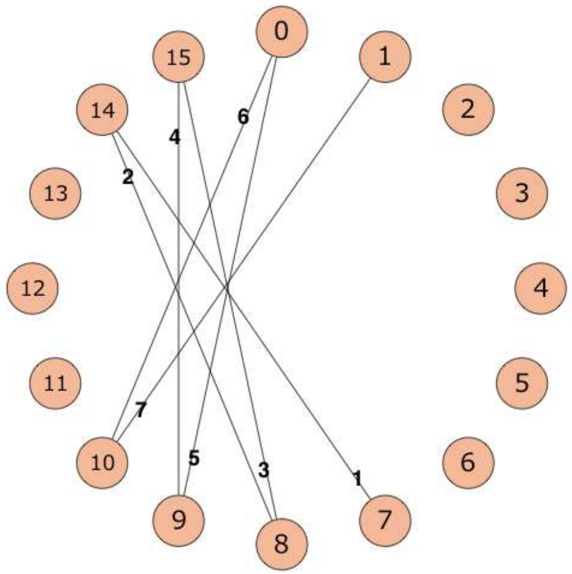
Problem G
Zigzag MST

CODE FESTIVAL 2016 Final

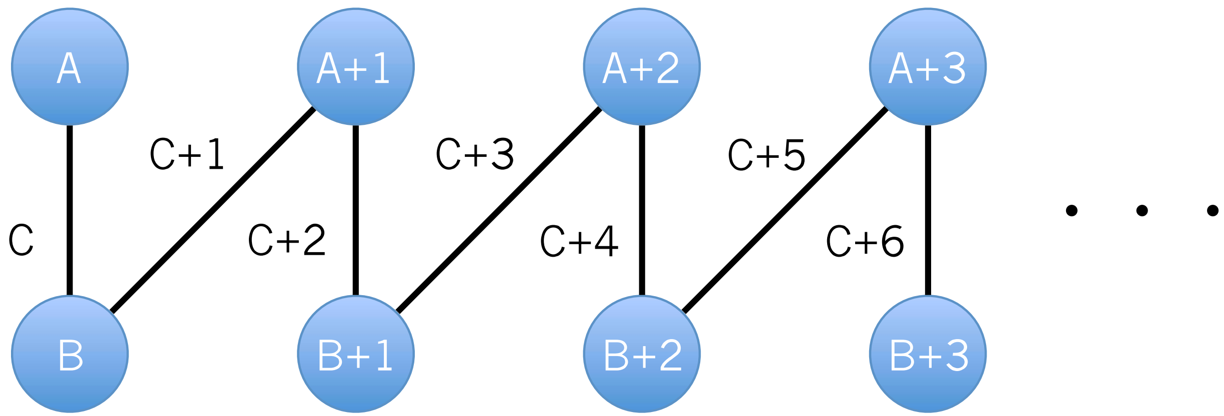
Problem

- There are N vertices
- Q queries will be processed
  - The i-th query: given A_i, B_i, C_i:
    - connect vertex A_i and B_i with an edge of cost C_i
    - connect vertex B_i and A_i+1 with an edge of cost C_i+1
    - connect vertex A_i+1 and B_i+1 with an edge of cost C_i+2
    - connect vertex B_i+1 and A_i+2 with an edge of cost C_i+3
    - …
- After all the queries are processed, find the weight of the minimum spanning tree (MST) of the graph.
- Constraints
  - $2 \leqq N \leqq 200{,}000$
  - $1 \leqq Q \leqq 200{,}000$
  - $1 \leqq C_i \leqq 10^9$



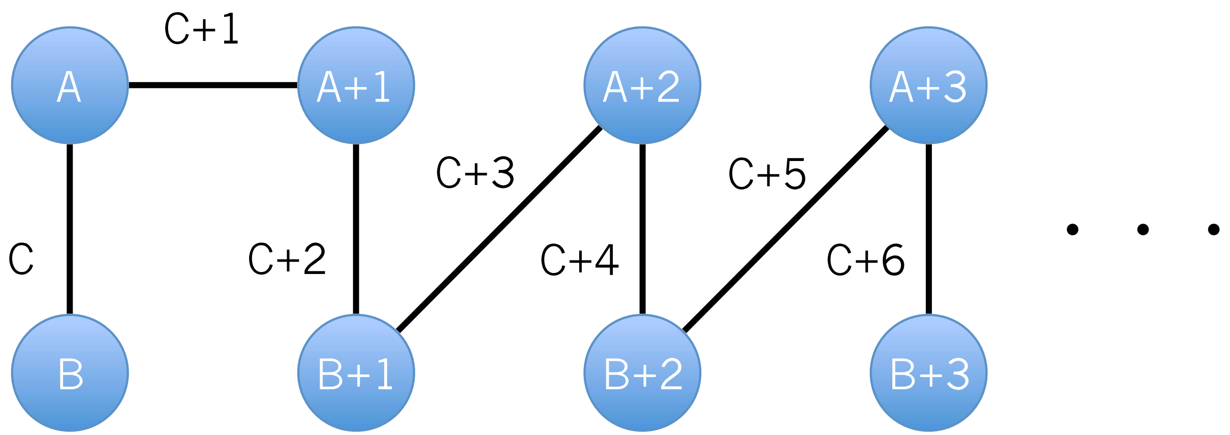
Solution

- Too many edges to directly find the MST
- We will examine the query
  - Let us apply Kruskal’s algorithm to find the MST
    - On the graph below, the edges are taken into account from left to right



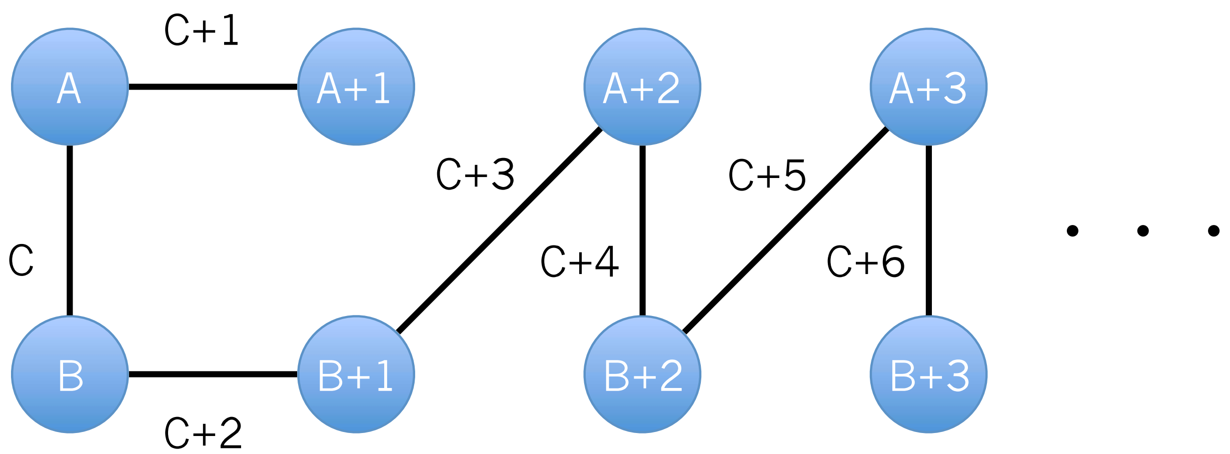
Solution

- We will examine the query
  - Let us apply Kruskal's algorithm to find the MST
  - Actually, we can relocate an edge as below without affecting the weight of MST!
    - When the edge with cost C+1 is taken into account, the edge with cost C must already be taken into account and vertices A and B must already be connected



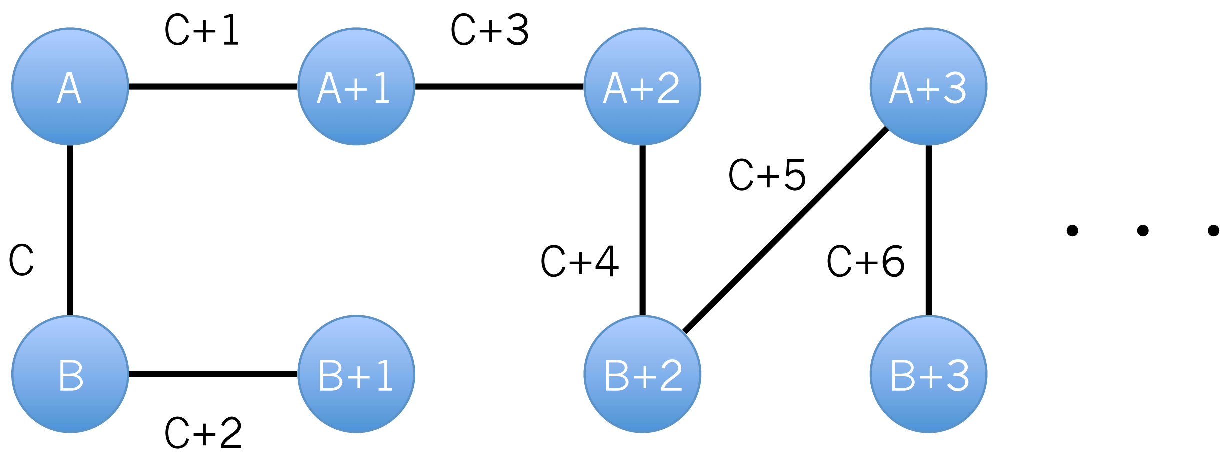
Solution

- We will examine the query
  - Relocating edges in the same way



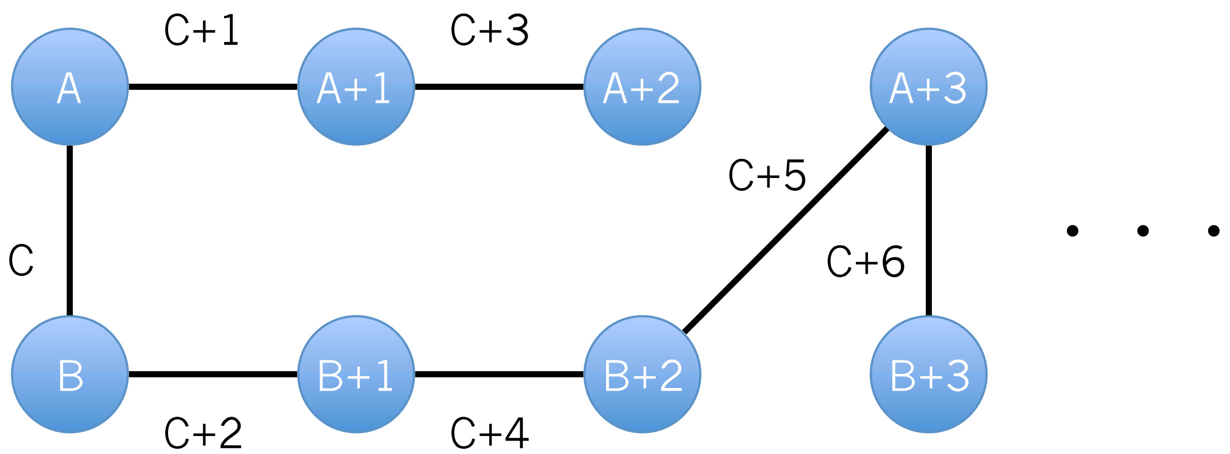
Solution

- We will examine the query
  - Relocating edges in the same way



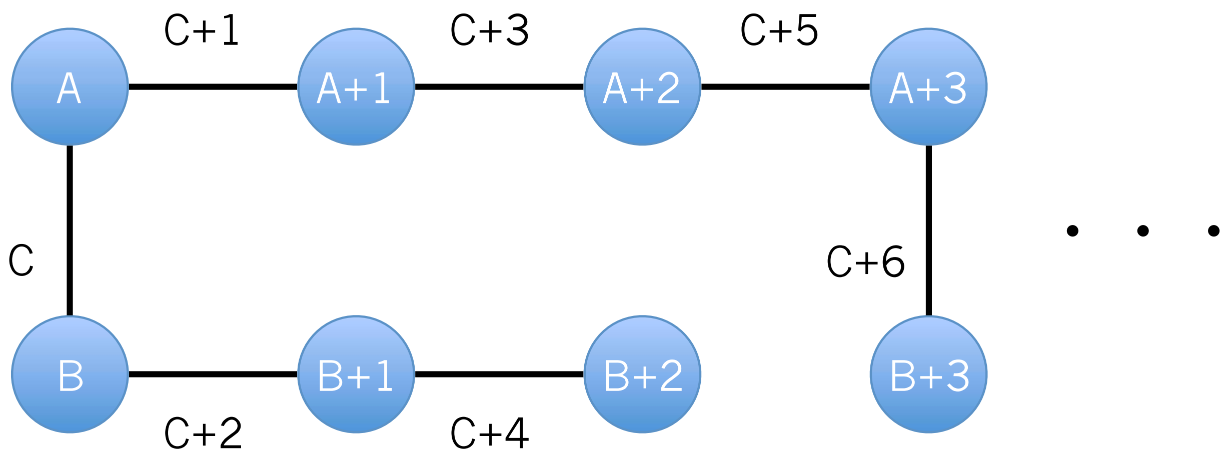
Solution

- We will examine the query
  - Relocating edges in the same way



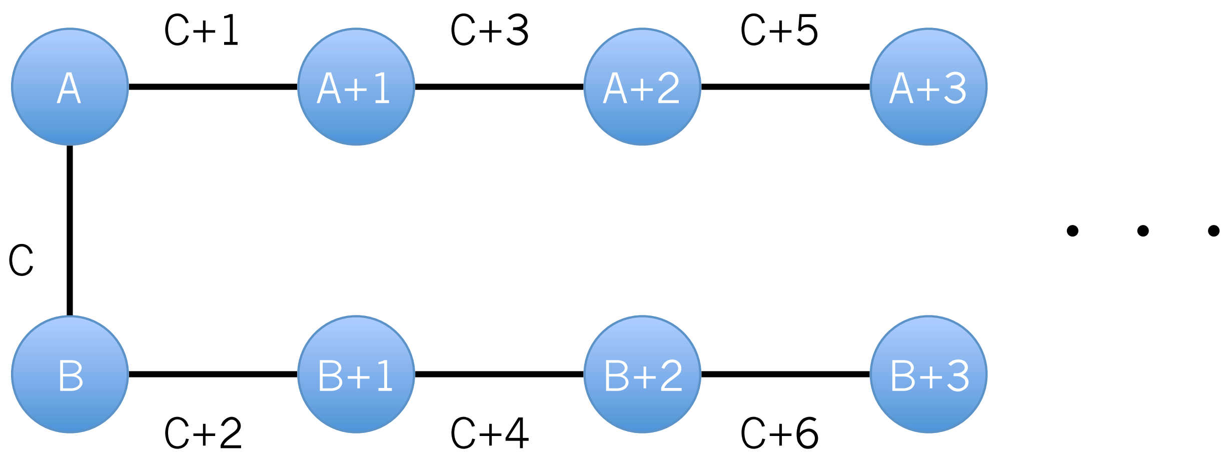
Solution

- We will examine the query
  - Relocating edges in the same way



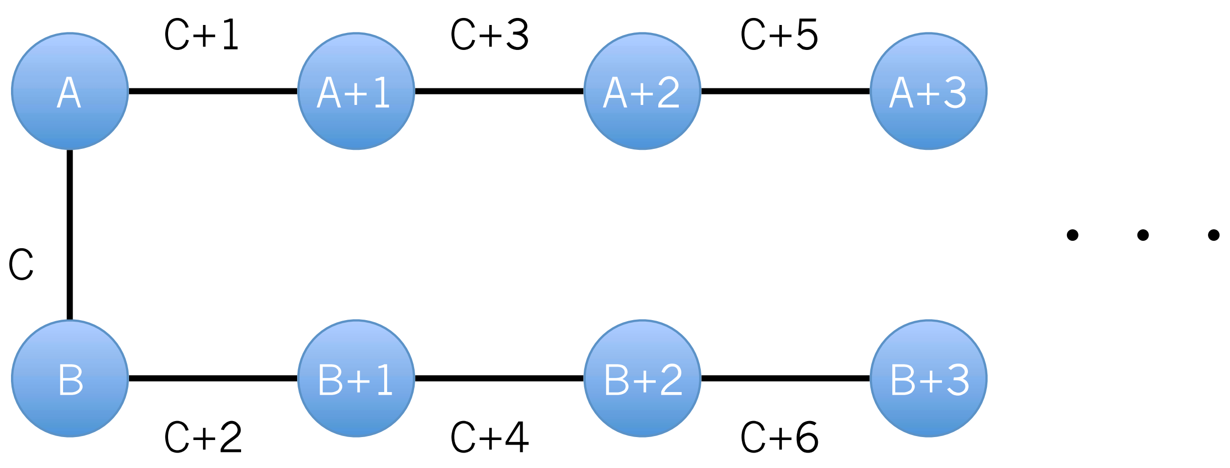
Solution

- We will examine the query
  - Relocating edges in the same way



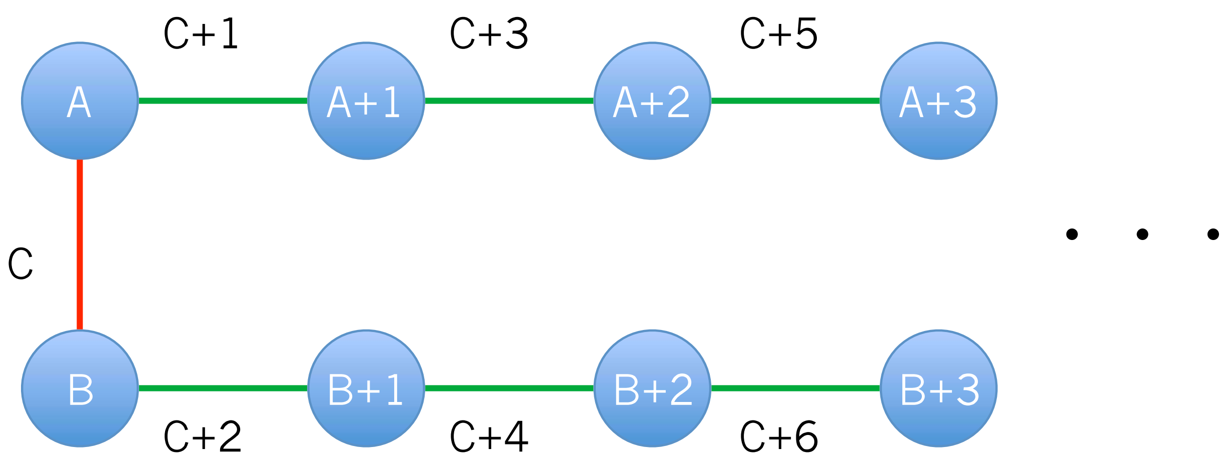
Solution

- After relocation, the edges can be classified into:
  - An edge of cost C connecting vertices A and B
  - Edges of cost C+1+2i connecting vertices A+i and A+1+i
  - Edges of cost C+2+2i connecting vertices B+i and B+1+i



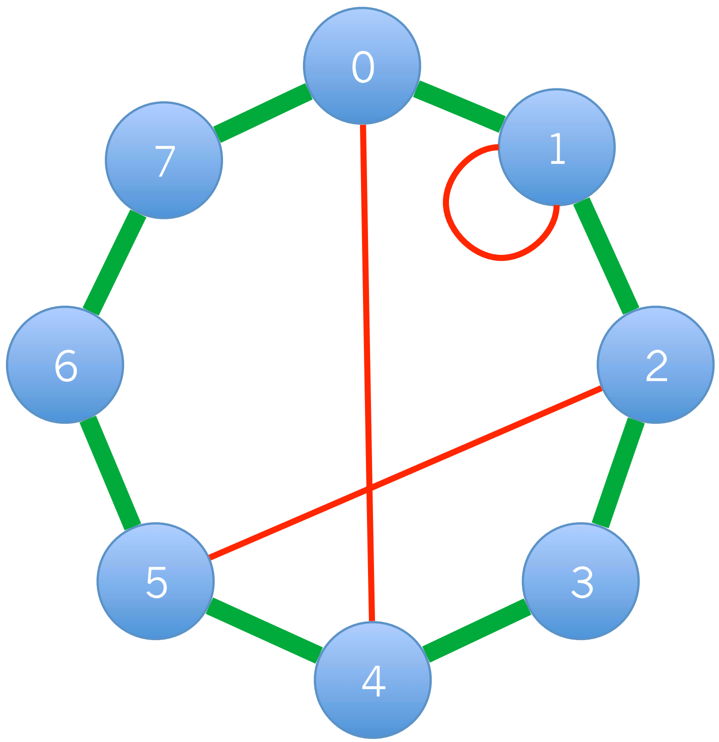
Solution

- After relocation, the edges can be classified into:
  - An edge of cost C connecting vertices A and B
  - Edges of cost C+1+2i connecting vertices A+i and A+1+i
  - Edges of cost C+2+2i connecting vertices B+i and B+1+i
- These types of edges are colored differently for illustrative purposes



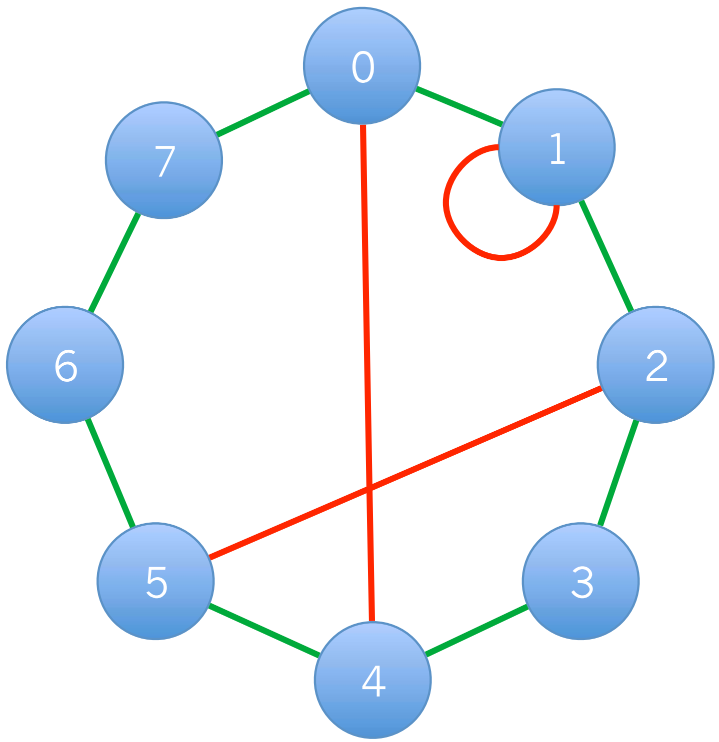
Solution

- After all the queries are processed and the edges are relocated, the graph looks as below:
  - There are infinitely many green edges where shown in green



Solution

- Among the green edges where shown in green, we can remove all but the one with the minimum weight, without affecting the weight of the MST
- Now there are only Q+N edges and we can simply find the MST



Solution

- How to find the green edge with the minimum weight where shown in green?
  - Green edges: edges with cost X+2i connecting vertices S+i and S+1+i
  - For simplicity, let us assume that green edges are spanned as follows:
    - First, connect vertices S and S+1 with an edge of cost X
    - From there, proceed clockwise spanning edges, while increasing the cost of an edge by 2 after each spanning
  - The algorithm
    - Output: c[i] = the minimum cost of an edge connecting vertices i and i+1

    1. Initialize each c[i] to ∞
    2. For each pair (S,X), perform an update: c[S] = min(c[S], X)
    3. For each i from 0 through N-1, perform an update: c[i+1] = min(c[i+1], c[i]+2). Execute this loop twice.
       - We are executing the loop twice to reflect the connection between N-1 and 0

Solution

- The time complexity
  - Finding the green edge with the minimum weight where shown in green: O(Q+N) in total
  - Finding MST afterwards: O((Q+N) log (Q+N))