G - ギガ国の道路事情

GIGACODE 2019 解説

問題概要

- N 頂点 M 辺の連結無向重み無しグラフが与えられます。
- $dist(i,j)$ を頂点 i と頂点 j の間の最短距離とします。

- そのとき、すべての頂点対 (a,b) における $dist(a,b)$ の総和を求めてください。

- $M - N \leq 777$（グラフの辺の数は頂点数とほぼ同じ）

小課題 1 (9 点)

- $N \leq 100$

- ワーシャルフロイド法を用いると、全頂点対最短距離が求まる
- 計算量は $O(N^3)$、普通に間に合います

小課題 2 (7 点 / 累積 16 点)

- $N \leq 4000$

- 各頂点 $1, 2, 3, \ldots, N$ をスタートとしたときの、全頂点への最短距離を求めることを考えます
- すべての辺の重みが 1 なので、queue を用いた幅優先探索で最短距離を $O(M)$ で求めることができます、よって計算量は $O(NM)$ です

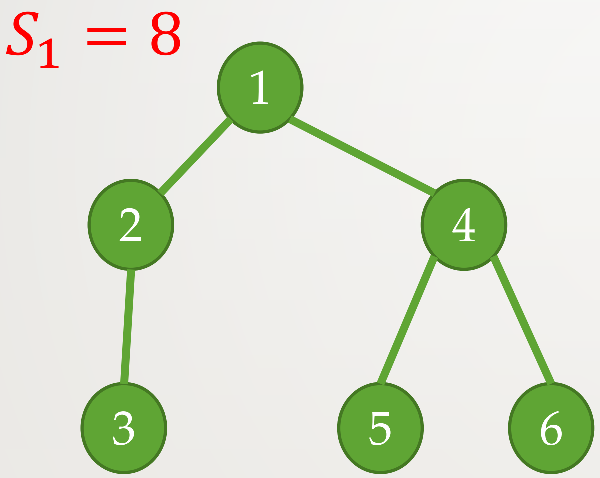
小課題 3 (12 点 / 累積 28 点)

- グラフは木である

- 頂点 i について、$dist(i,1) + dist(i,2) + \cdots + dist(i,N) = S_i$ とします。
- まず、S_1 の値は幅優先探索をすることによって $O(N)$ で求まります。

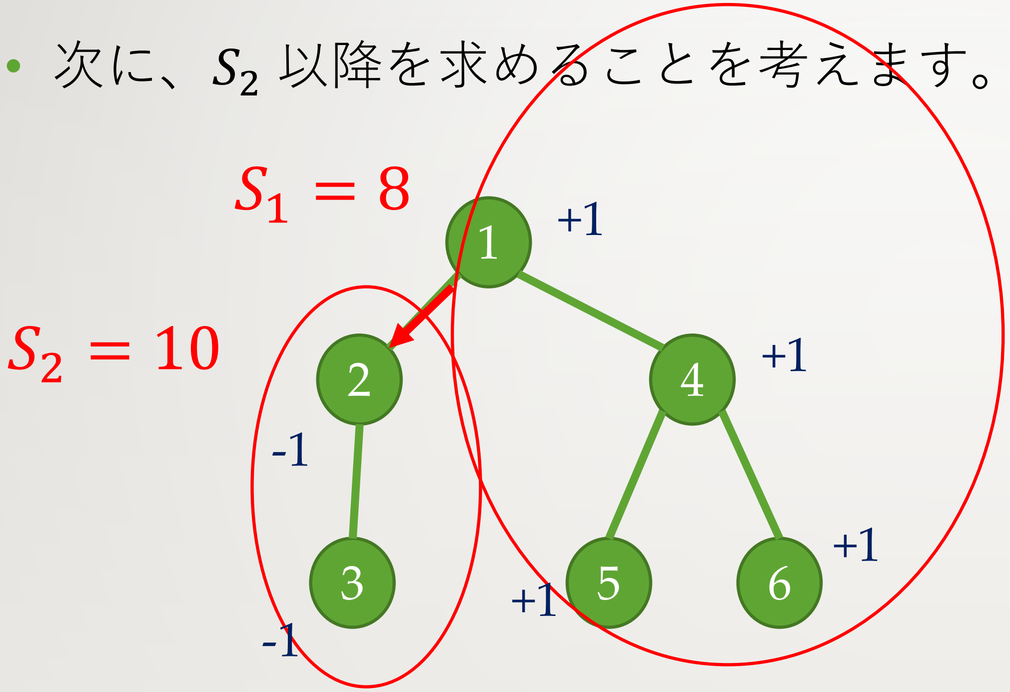
小課題 3 (12 点 / 累積 28 点)

- 次に、S_2 以降を求めることを考えます。



小課題 3 (12 点 / 累積 28 点)

- 次に、S_2 以降を求めることを考えます。

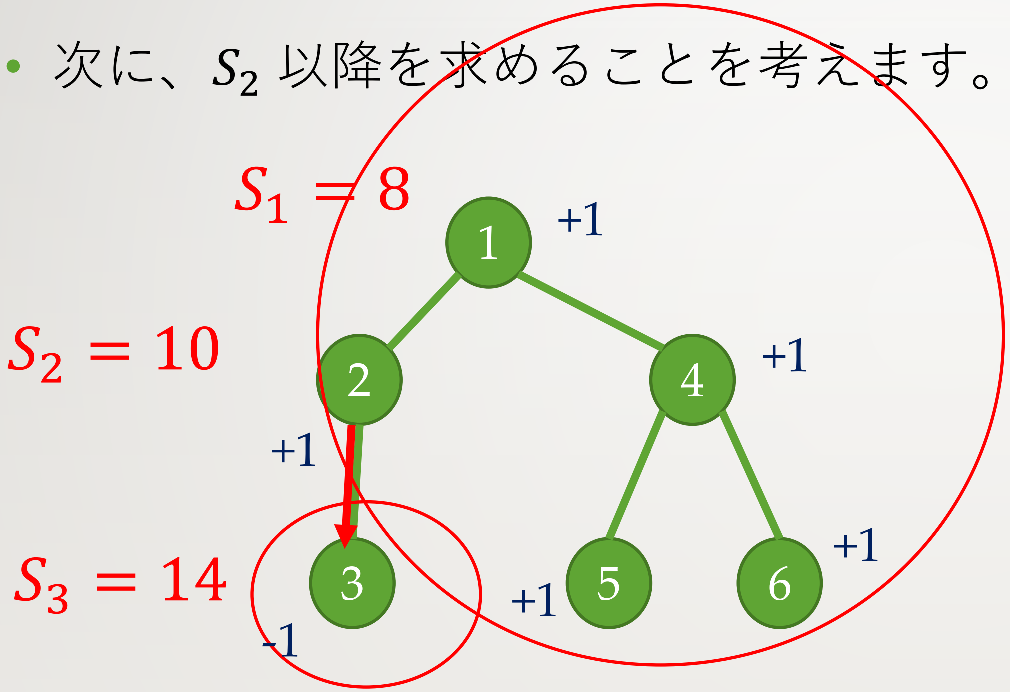


部分木のサイズがそれぞれ 2, 4 なので、

$$S_2 = S_1 + 4 - 2$$

小課題 3 (12 点 / 累積 28 点)

- 次に、S_2 以降を求めることを考えます。

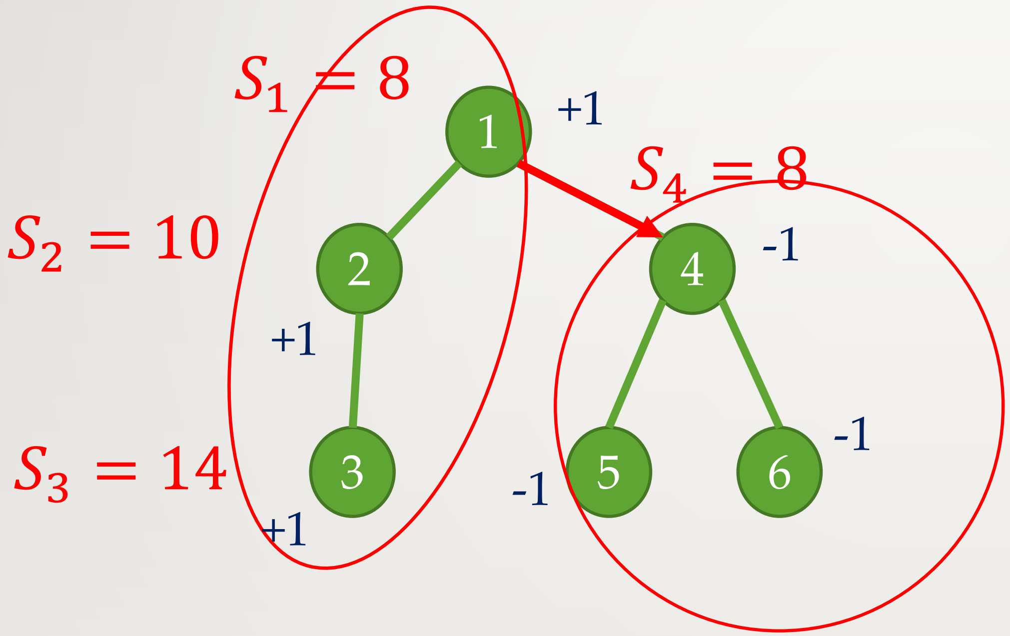


部分木のサイズがそれぞれ 1, 5 なので、

$$S_3 = S_2 + 5 - 1$$

小課題 3 (12 点 / 累積 28 点)

- 次に、S_2 以降を求めることを考えます。

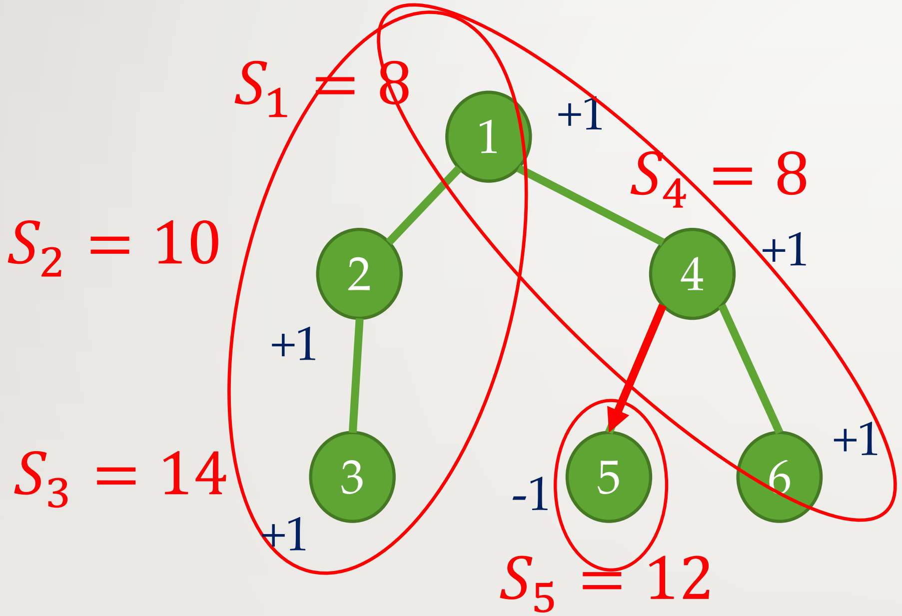


部分木のサイズがそれぞれ 3, 3 なので、

$$S_4 = S_1 + 3 - 3$$

小課題 3 (12 点 / 累積 28 点)

- 次に、S_2 以降を求めることを考えます。

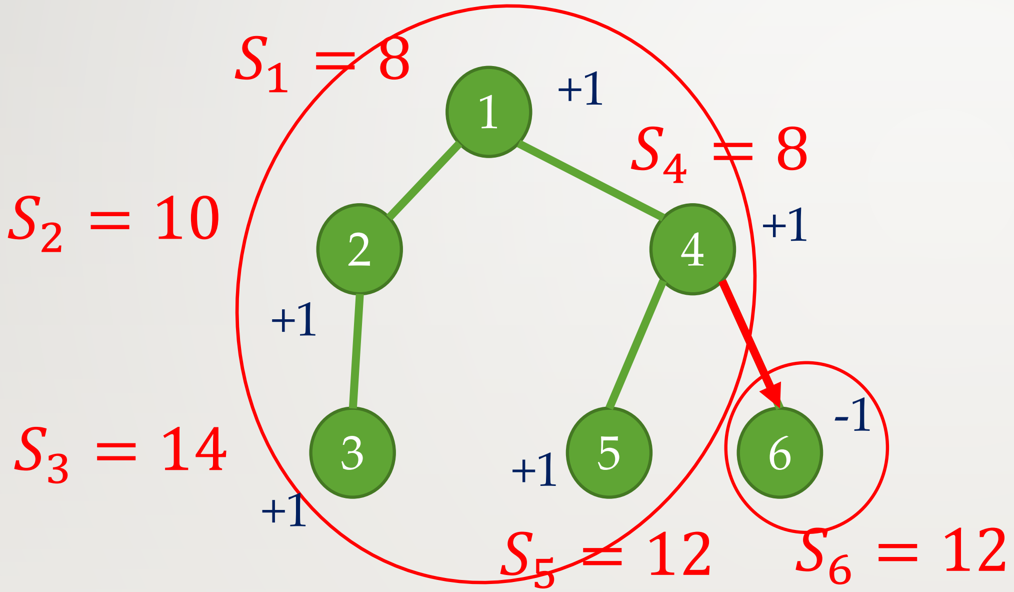


部分木のサイズがそれぞれ 1, 5 なので、

$$S_5 = S_4 + 5 - 1$$

小課題 3 (12 点 / 累積 28 点)

- 次に、S_2 以降を求めることを考えます。



部分木のサイズがそれぞれ 1, 5 なので、

$$S_6 = S_4 + 5 - 1$$

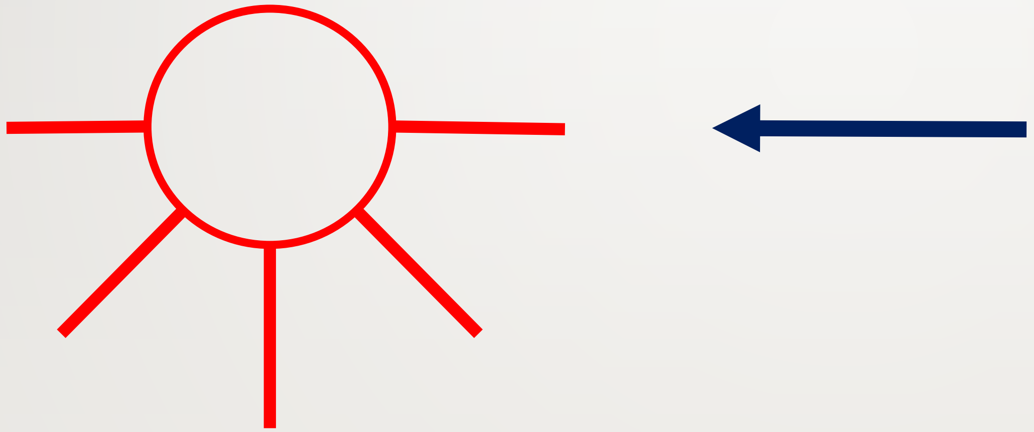
小課題 3 (12 点 / 累積 28 点)

- このように、DFS をするとすべての S_i が分かります。

- あとは、答えは $\frac{S_1+S_2+S_3+S_4+\cdots+S_N}{2}$ となります。
- ここまで実装すると、28 点が取れます。

小課題 4 (13 点 / 累積 41 点)

- $N = M$ です。
- 木から 1 本追加したものなので、1 個のループができます。



$N = N$ の場合のグラフの概形

小課題 4 (13 点 / 累積 41 点)

- あとは、
  - ループ内頂点同士
  - ループ内頂点からループ外頂点
  - ループ外頂点同士

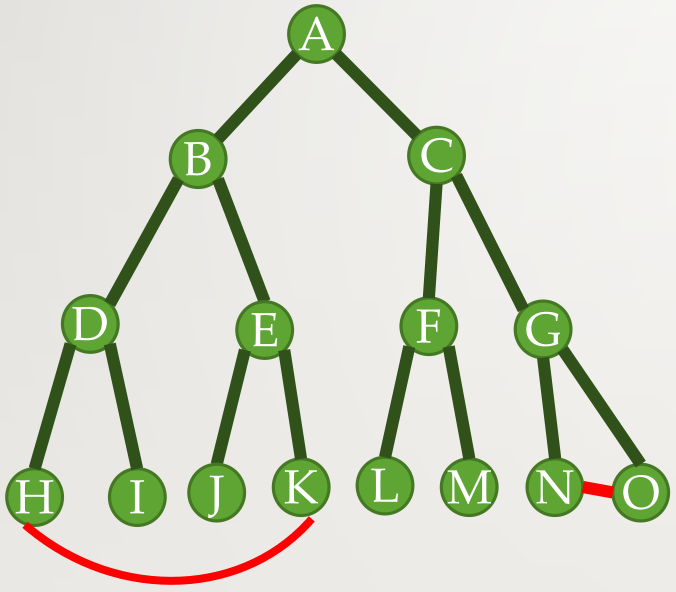
- の 3 つに場合分けすると、木の場合と同様に解くことができます。

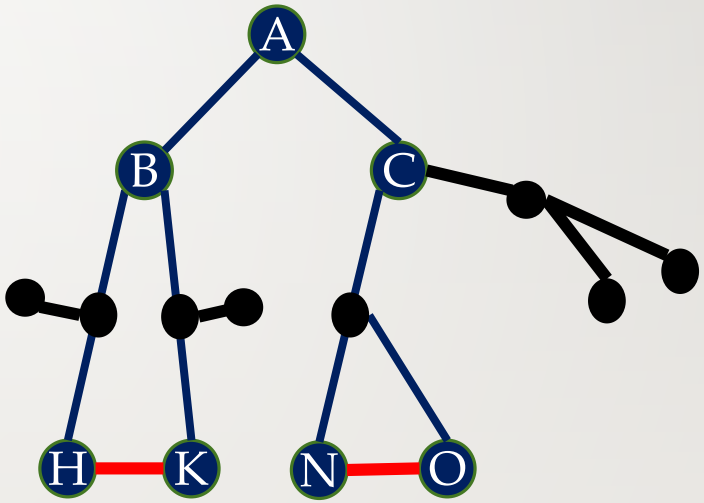
満点解法への道

- $M - N \leq 777$
- これは全域木に高々 778 本を足したものです
- グラフを圧縮できないか！？

満点解法への道

- 例えば以下のグラフを考える





満点解法への道

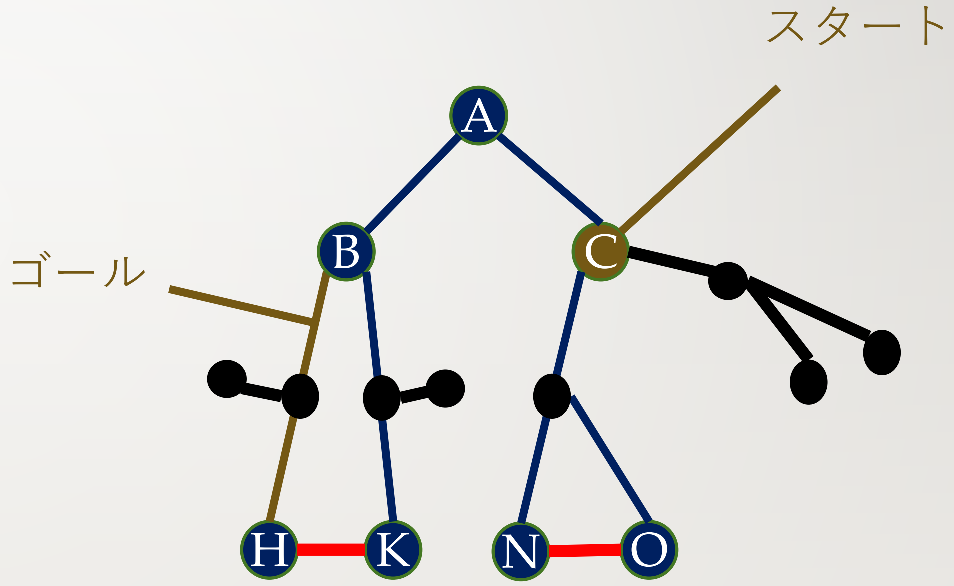
- そのように、木を圧縮すると、高々 4(M-N) 頂点しか考えなくてよくなる
  - 補足：DFS 木を作って上手くやると 3(M-N) 頂点でよくなる

- 取り敢えずこの 4(M-N) あるいは 3(M-N) 頂点を「重要な頂点」と表す

- 全重要な頂点対の最短距離は、ダイクストラ法を用いて、$O((M-N)^2 \log(M-N))$ で求まる

満点解法への道

- あとは、ある頂点 x から辺 (u, v) 上の頂点の合計距離を考える

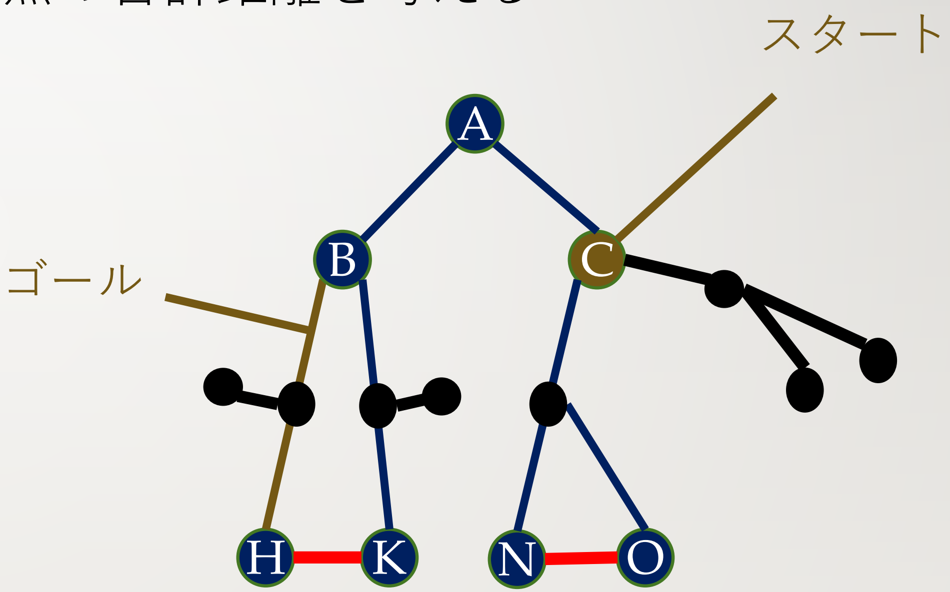


満点解法への道

- あとは、ある頂点 x から辺 (u, v) 上の頂点の合計距離を考える

(a) x → u → 辺上の頂点
(b) x → v → 辺上の頂点

(a) と (b) どちらが近いかは、
ある境界で分かれる

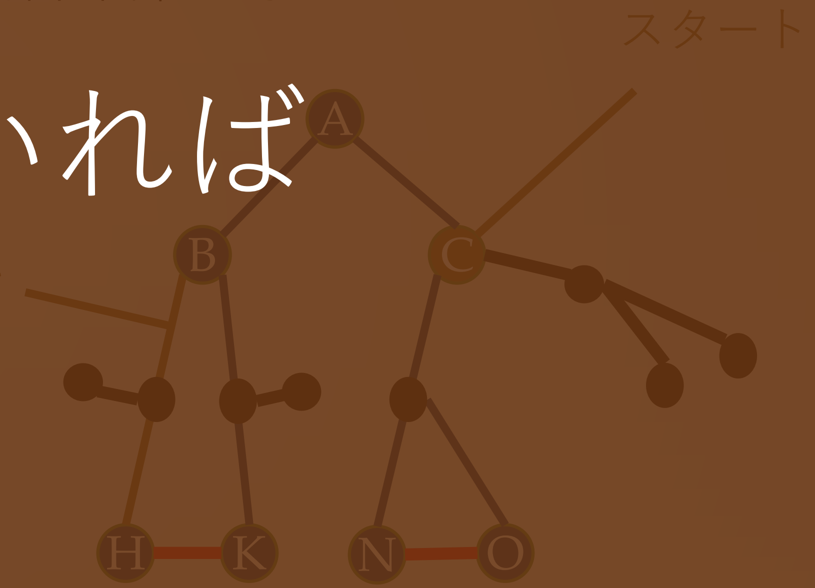


満点解法への道

- あとは、ある頂点 x から辺 (u, v) 上の頂点の合計距離を考える

(a) x → u → 辺上の頂点
(b) x → v → 辺上の頂点

(a) と (b) どちらが近いかは、
ある境界で分かれる



累積和を用いれば
解ける

満点解法

- $O(N(M-N))$ 回、頂点から辺上頂点の距離の合計を求める必要がある

- 境界を求めるのに二分探索を使った場合
  - $O(N(M-N)logN)$ → 91 点

- 境界を求めるのに前計算を使った場合
  - $O\left(N(M-N)\right)$ → 100 点