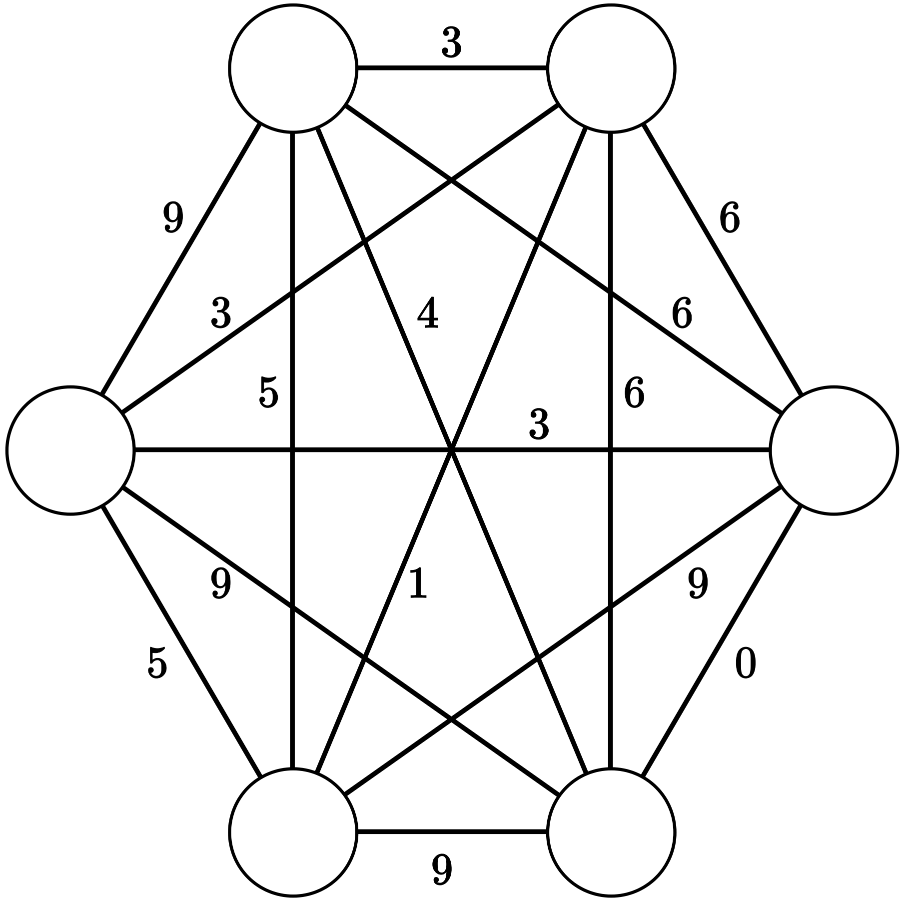
JOI 2025/2026 ファイナルステージ Day 1 – マルチコミュニケーション 2 (Multi Communication 2) 解説

解説担当: 山縣 龍人 (tatyam)

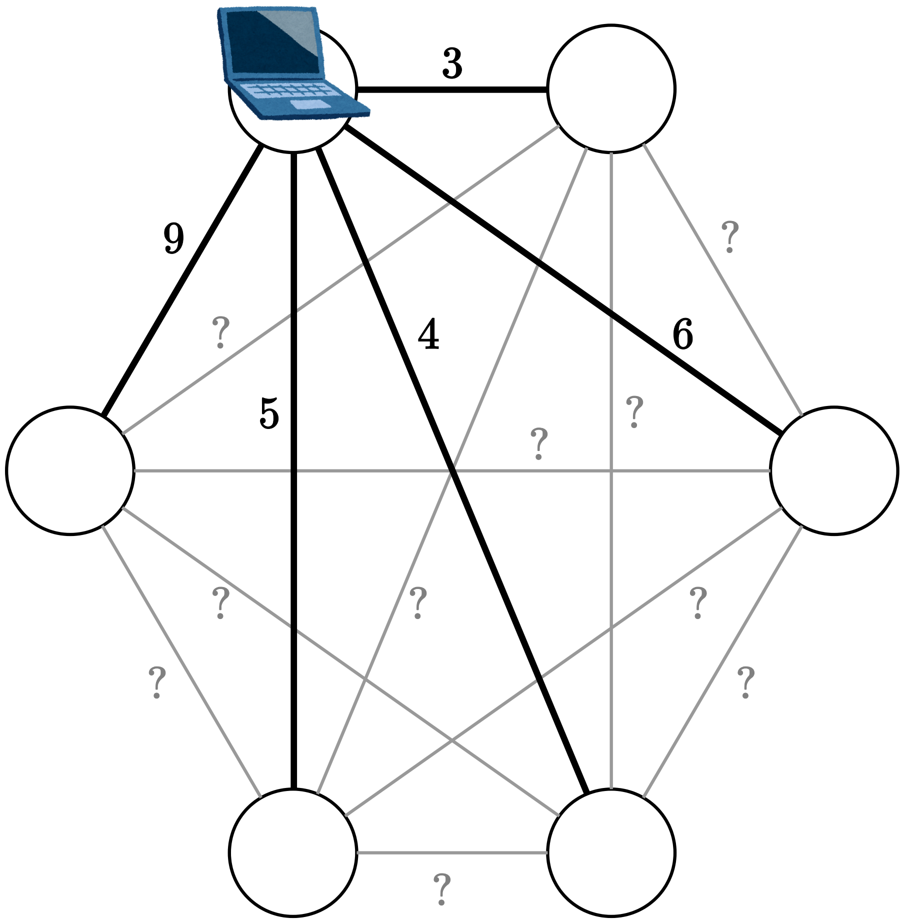
問題概要

- N 頂点の重み付き完全無向グラフがある.



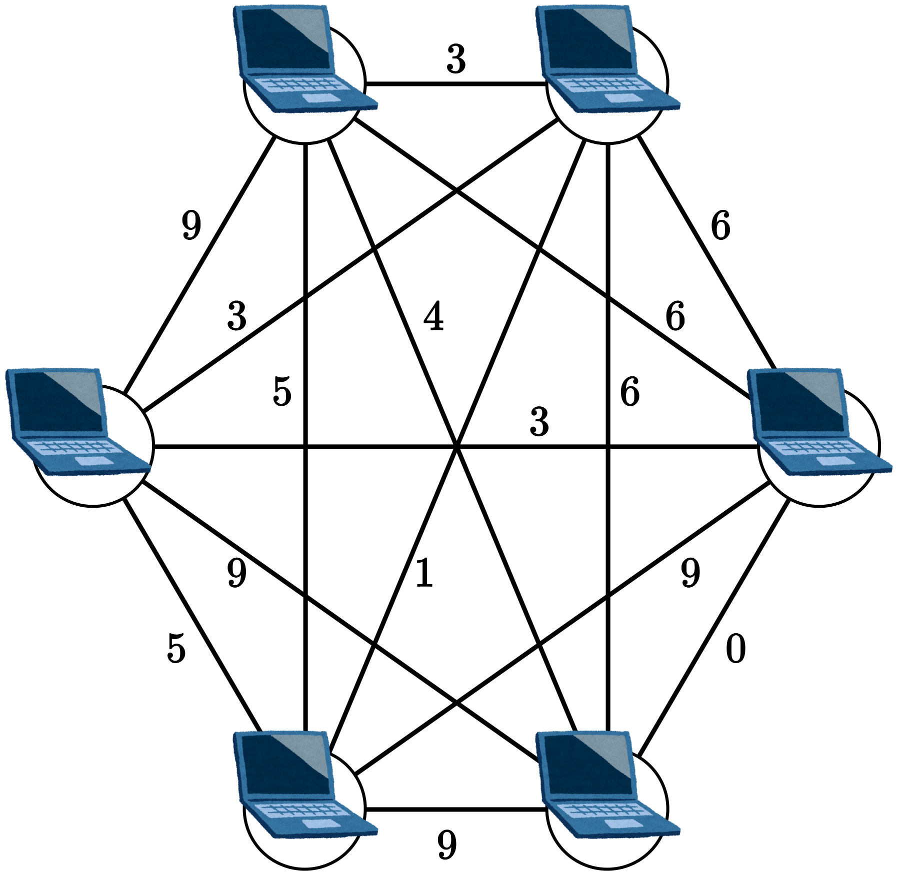
問題概要

- N 頂点の重み付き完全無向グラフがあるが，重みは隠されている．
- 各頂点は，その頂点に隣接する辺の重みが分かる．



問題概要

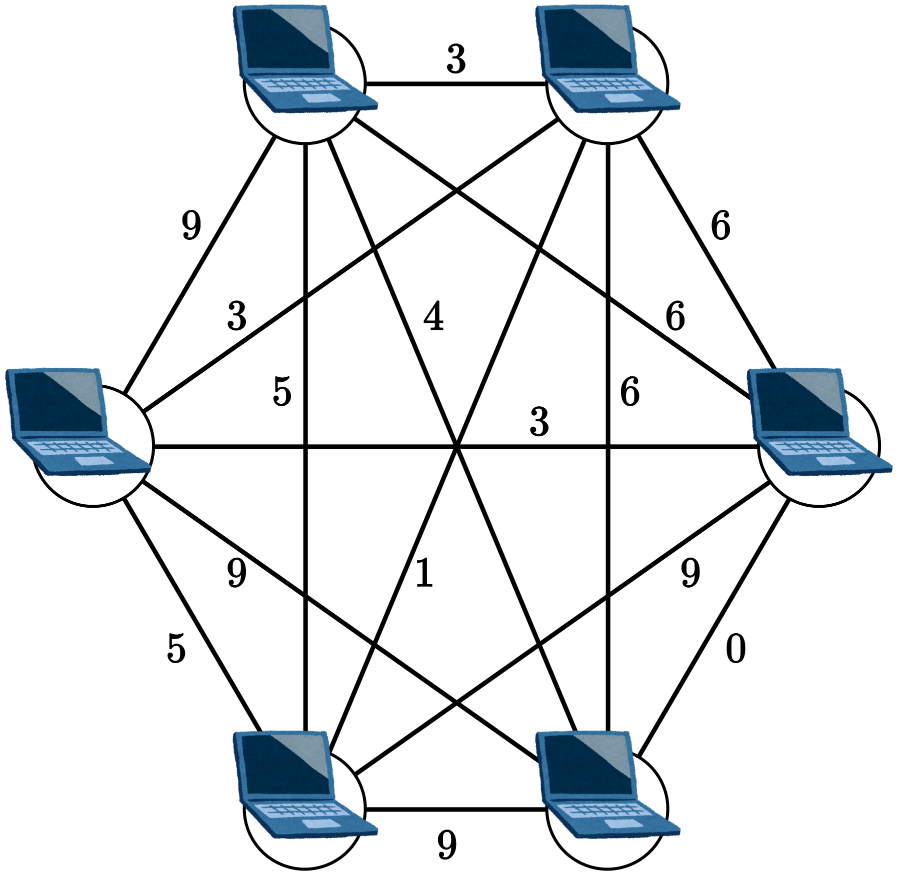
- N 頂点の重み付き完全無向グラフがあるが，重みは隠されている.
- 各頂点は，その頂点に隣接する辺の重みが分かる.
- 全頂点が協力して **分散計算** を行い，最小全域木の重み X を求める.



分散計算の方法

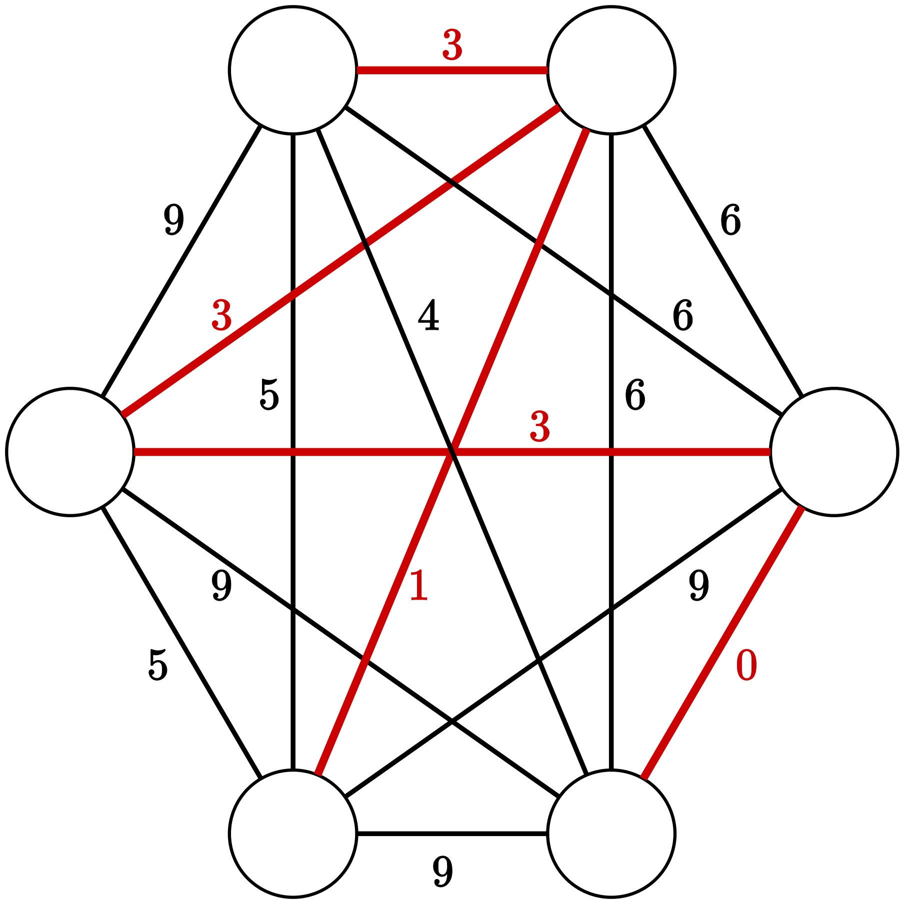
以下の **ラウンド** を繰り返す.

1. 前のラウンドの各頂点から 64 bit の情報を受け取る
2. 最小全域木の重み X が求められたなら，それを回答する.
3. そうでなければ，次のラウンドの各頂点へ 64 bit の情報を送信する.
   (次のラウンドの自分自身へも 64 bit の情報しか引き継げない！)



問題概要

- N 頂点の重み付き完全無向グラフがあるが，重みは隠されている.
- 各頂点は，その頂点に隣接する辺の重みが分かる.
- 全頂点が協力して **分散計算** を行い，最小全域木の重み X を求める.
- 何ラウンドかかるか？
- $N \leq 256$，辺重み $A_{i,j}$ は 48 bit



小課題

	配点	$N \leq$	$A_{i,j}$ は	満点には
1	5 点	64	1 bit	6 ラウンド
2	10 点	256	1 bit	6 ラウンド
3	15 点	64	48 bit	80 ラウンド
4	40 点	256	20 bit	6 ラウンド
5	15 点	256	40 bit	6 ラウンド
6	15 点	256	48 bit	6 ラウンド

小課題内の点数
(小課題 3 を除く)

- 6 ラウンド $\to$ 100%
- 7 ラウンド $\to$ 80%
- 8 ラウンド $\to$ 60%
- 9 ラウンド $\to$ 40%
- 10 ラウンド $\to$ 30%

小課題

	配点	$N \leq$	$A_{i,j}$ は	満点には
1	5 点	64	1 bit	6 ラウンド
2	10 点	256	1 bit	6 ラウンド
3	15 点	64	48 bit	80 ラウンド
4	40 点	256	20 bit	6 ラウンド
5	15 点	256	40 bit	6 ラウンド
6	15 点	256	48 bit	6 ラウンド

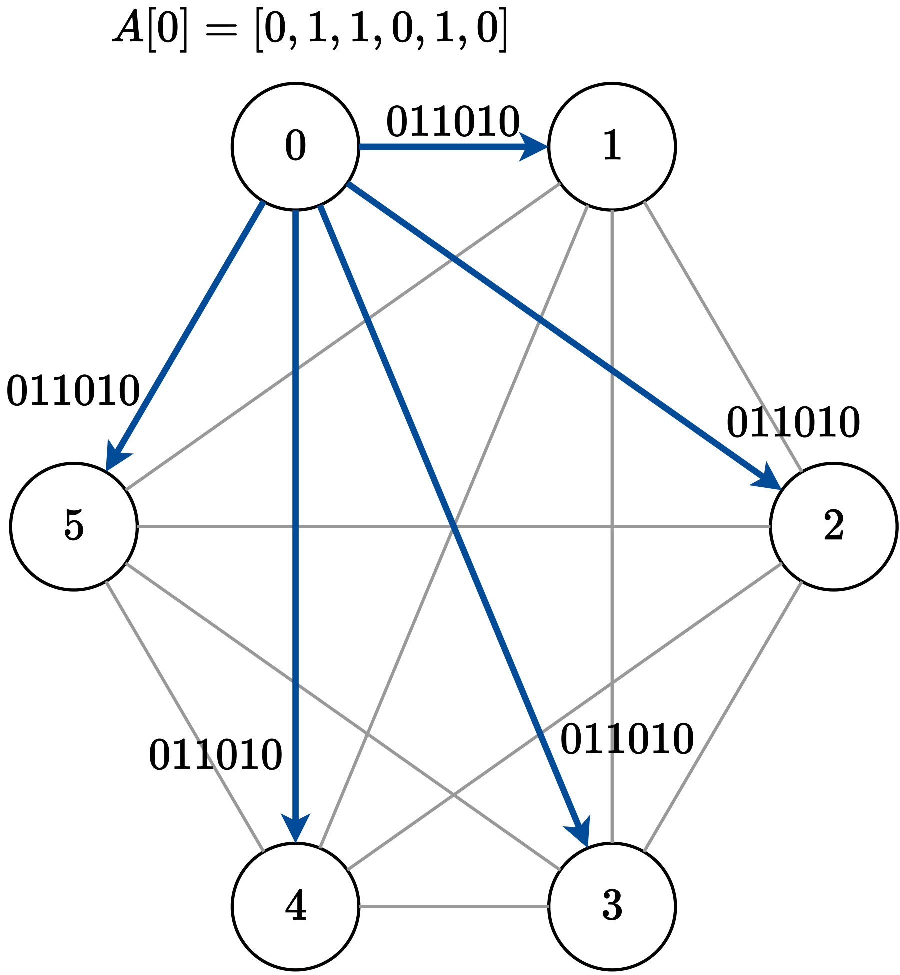
小課題内の点数 (小課題 3 を除く)

- 6 ラウンド $\to$ 100%
- 7 ラウンド $\to$ 80%
- 8 ラウンド $\to$ 60%
- 9 ラウンド $\to$ 40%
- 10 ラウンド $\to$ 30%

6 ラウンド？🤔　本当にそんなことが可能なのか？🤔🤔

小課題 1 : $N \leq 64$, $A_{i,j}$ は 1 bit

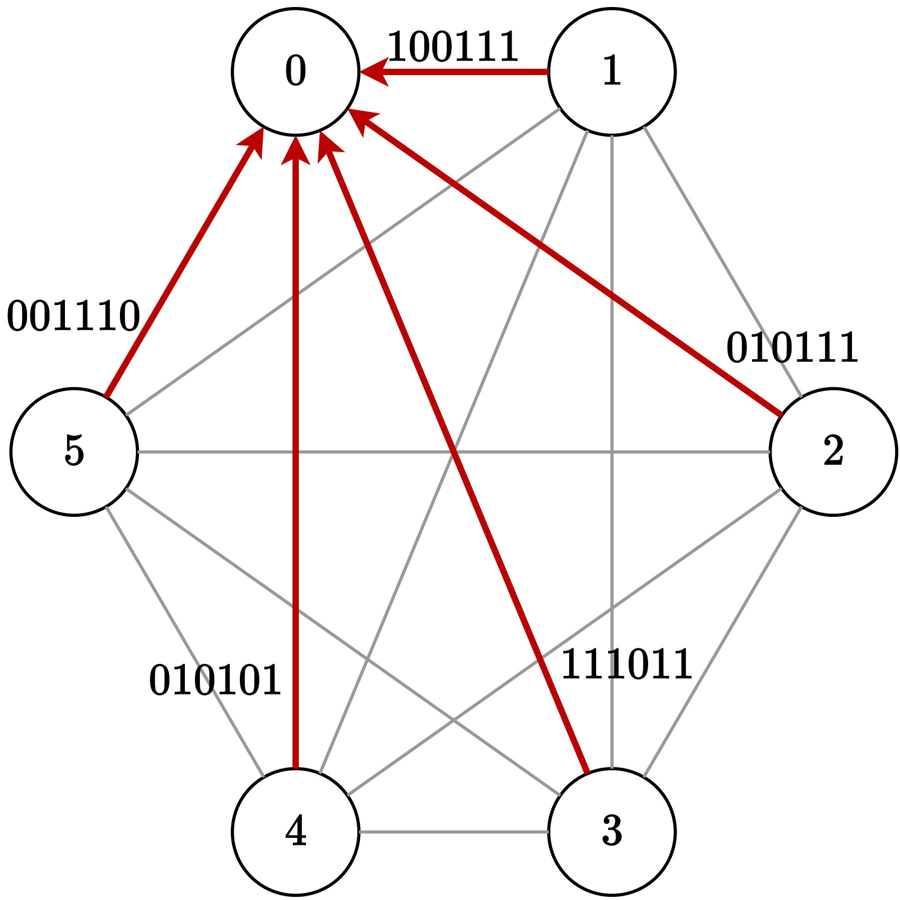
- 0 ラウンド目で，各頂点 i は，行 $A[i]$ の情報を 64 bit に詰めてすべての頂点に放送することができる！
  - 全頂点に送ることを **放送** と呼ぶことにします．



小課題 1 : $N \le 64$, $A_{i,j}$ は 1 bit

- 0 ラウンド目で，各頂点 i は，行 $A[i]$ の情報を 64 bit に詰めてすべての頂点に放送することができる！
  - 全頂点に送ることを **放送** と呼ぶことにします.
- 1 ラウンド目ですべての辺の情報が集まるので，最小全域木を計算する.

2 ラウンドでできた！



小課題 2 : $N \leq 256$, $A_{i,j}$ は 1 bit

- 行 $A[i]$ の情報を 64 bit に詰められない… → **分けて送ろう！**
- 分けて送ろうにも，送られた情報をすべて覚えておくことができない…

どうする？

小課題 2 : $N \leq 256,\ A_{i,j}$ は 1 bit

- 行 $A[i]$ の情報を 64 bit に詰められない… → **分けて送ろう！**
- 分けて送ろうにも，送られた情報をすべて覚えておくことができない

→ 送られた情報のうち，重み 0 の辺をすべて採用して，**現在の連結成分の状態を覚えておけば良い！**

連結成分の状態を覚えておく

連結成分の状態を覚えておくには…

→ UnionFind を再現しよう！

UnionFind を再現しよう！

- $\mathrm{leader}[i] :=$ 頂点 i を含む連結成分の代表元 (例えば, 連結成分の中で最小の頂点)

の配列が分かっていれば良い.

ラウンドの終わりに頂点 i が $\mathrm{leader}[i]$ を放送すれば，全頂点に配列 leader が渡る！

UnionFind を再現しよう！

- $\mathrm{leader}[i]$:= 頂点 i を含む連結成分の代表元　の配列が分かっていれば良い.

ラウンドの終わりに頂点 i が $\mathrm{leader}[i]$ を放送すれば，全頂点に配列 leader が渡る！

注意点：正しい $\mathrm{leader}[i]$ を放送するためには,

- 全員が同じように辺の情報を受け取り,
- 全員が同じように辺を採用し,
- 全員が同じように leader を更新する

必要がある！

小課題 2 : $N \le 256$, $A_{i,j}$ は 1 bit

- 各頂点 i は行 $A[i]$ の情報を 5 **回に分けて** 52 bit **ずつ放送する.**
- また，毎回 $\mathrm{leader}[i]$ を放送する (8 bit).
- 全員が同じように辺の情報を受け取り，
- 重みが 0 である辺を (サイクルができなければ) すべて採用し，
- 全員が同じように leader を更新する.
- 最終的な連結成分数から答えを逆算

5 回の放送 → 6 ラウンドでできた！

テクニック (1) : 採用された辺の情報を持たせる

実は,

- $\mathrm{leader}[i] :=$ 頂点 i を含む連結成分の代表元 (例えば, 連結成分の中で最小の頂点)

の配列に, 「どの辺が採用されたか？」を持たせることができる！

テクニック (1) : 採用された辺の情報を持たせる

現在採用されている辺によって作られる森を T とする.

- $\mathrm{leader}[i]$:= 頂点 i を含む連結成分の代表元 (例えば, 連結成分の中で最小の頂点)

の代わりに,

- $\mathrm{parent}[i]$:= T 上で頂点 i から $\mathrm{leader}[i]$ 方向へ 1 つ進んだ頂点

を持つ (T を有向森として表現している).
parent から leader は復元できるので，情報が追加されている！☺️

小課題 $3: N \leq 64$, $A_{i,j}$ は $48\,\mathrm{bit}$, 80 ラウンドで満点

最小全域木のアルゴリズムといえば…？

小課題 $3: N \leq 64,\ A_{i,j}$ は $48\,\mathrm{bit},\ 80$ ラウンドで満点

最小全域木のアルゴリズムといえば…？

- クラスカル法 (いつもの)
- プリム法 (ダイクストラ法みたいな)
- ブルーフカ法 (？)

小課題 $3: N \le 64,\ A_{i,j}$ は 48 bit，80 ラウンドで満点

最小全域木のアルゴリズムといえば…？

- クラスカル法
- **プリム法** ← 小課題 3 の正解
- ブルーフカ法

クラスカル法でも可能 (ブルーフカ法に似るので省略)

最小全域木の性質

定理

頂点集合を V とする.
任意の $S \subset V$ に対して, $T := V \setminus S$ とする.
$S - T$ 間の最小重みの辺は, 最小全域木に採用して良い.

プリム法とは？

以下を $N-1$ 回繰り返す：

- 頂点 0 と連結な頂点の集合を S，そうでない頂点の集合を T とする．はじめ，$S=\{0\}, T=\{1,2,\ldots,N-1\}$ である．
- $S-T$ 間の辺 $u-v$ であって，重みが最小のものを選ぶ．
- 辺 $u-v$ を採用し，v を T から S に移動させる．

小課題 3 : プリム法

- $i \in T$ である各頂点 i は，$i - S$ 間の最小重みを放送する ($48\,\mathrm{bit}$).
  → この重みが最小である i を S に加える.
- 上記を放送するために，S, T の要素が分かる必要がある
  → 各頂点は $i \in S$ または $i \in T$ の $1\,\mathrm{bit}$ を放送する.
- 採用された辺重みを合計する必要がある
  → 頂点 0 は，採用された辺重みの合計を放送する ($56\,\mathrm{bit}$).

N ラウンドでできた！

ブルーフカ法とは？

以下を高々 $\mathrm{floor}(\log_2(N))$ 回繰り返す：

1. 各連結成分について，その連結成分の内外をまたぐ辺のうち，重みが最小のものを求める.
2. それらを (サイクルにならない限り) 採用する.

1 フェーズにおいて，各連結成分は少なくとも 1 つの連結成分と連結になる.
→ k フェーズ行ったとき，各連結成分のサイズは 2^k 以上！

💡 テクニック (2) : ブルーフカ法豆知識

- 各連結成分のサイズが x 以上である，かつ
- $N \le 2x - 1$ である

$\implies$ グラフは連結である

余計に 1 フェーズやらないよう注意！

ブルーフカ法とは？

以下を高々 $\log_2(N)$ 回繰り返す：

1. 各連結成分について，その連結成分の内外をまたぐ辺のうち，重みが最小のものを求める.
2. それらを (サイクルにならない限り) 採用する.

辺を並列に探すことができる

→ 分散計算に適している！

ブルーフカ法 : 9 ラウンド

- 各頂点 i は，$\mathrm{leader}[i]$ を放送する (8 bit).
- 各頂点 i は，$i-$ (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).

どうやって採用された辺の重みを合計する？

ブルーフカ法 + 集計 : 10 ラウンド

- 各頂点 i は，$\mathrm{parent}[i]$ を放送する (8 bit).
- 各頂点 i は，i — (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).

8 回の放送で最小全域木を求める → 追加の 1 回の放送で,
最小全域木に採用された辺の重みを放送する (頂点 i は $A_{i,\mathrm{parent}[i]}$ を放送)

放送 9 回 → 10 ラウンドでできた！

ブルーフカ法 : 9 ラウンド

- 各頂点 i は, $\mathrm{leader}[i]$ を放送する (8 bit).
- 各頂点 i は, i − (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).

送る必要のない情報はないか？

テクニック (3) : 自分自身に送る情報を変更

- 各頂点 i は, $\mathrm{leader}[i]$ を放送する (8 bit).
- 各頂点 i は, $i-$ (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).
  - 自分自身にこの情報を送る必要はない！

テクニック (3) : 自分自身に送る情報を変更

- 各頂点 i は，$\mathrm{leader}[i]$ を放送する (8 bit).
- 各頂点 i は，i − (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).
    - **自分自身へは，「採用された辺の重みの合計」を送信する** (56 bit)

放送 8 回 → 9 ラウンドでできた！

一応こんな方法も…？

- 各頂点 i は，$\mathrm{leader}[i]$ を放送する (8 bit).
  - **種類が 8 bit もない！** 連結成分の番号を $0, 1, 2, \ldots$ に振り直しても問題ない！
- 各頂点 i は，i − (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).

一応こんな方法も…？

- 各頂点 i は，$\mathrm{leader}[i]$ を放送する ($\mathrm{ceil}(\log_2 N) - 1\,\mathrm{bit}$).
- 各頂点 i は，i − (i と連結でない頂点) 間の (最小重み, 行き先) を放送する ($47 + \mathrm{ceil}(\log_2 N)\,\mathrm{bit}$).
- $x := 18 - 2\mathrm{ceil}(\log_2 N)$ とする．各頂点 i は，「採用された辺の重みの合計」の 2^x 進法で i 桁目を放送する ($x\,\mathrm{bit}$).
  - $N \geq 4$ であれば足りる！

放送 8 回 → 9 ラウンドでできた！

テクニック (4) : 初手 3 倍

さっき 10 ラウンドになってしまったブルーフカ法ですが…？

- 各頂点 i は，$\textcolor{red}{\mathrm{parent}[i]}$ を放送する (8 bit).
- 各頂点 i は，$i-$ (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).
- 8 回の放送で最小全域木を求める → 追加の 1 回の放送で，
  最小全域木に採用された辺の重みを放送する

送る必要のない情報はないか？

テクニック (4) : 初手 3 倍

0 ラウンド目では…？

- 各頂点 i は，$\mathrm{parent}[i]$ を放送する (8 bit).
- 各頂点 i は，$i-(i$ と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).

テクニック (4) : 初手 3 倍

0 ラウンド目では…？

- ~~各頂点 i は，$\mathrm{parent}[i]$ を放送する (8 bit).~~
- 各頂点 i は，i − (i と連結でない頂点) 間の (~~最小重み~~, 行き先) を放送する (~~56 bit~~ 8 bit).

8 bit しか必要ない！

テクニック (4) : 初手 3 倍

0 ラウンド目では，もう 1 辺送れる！

- 各頂点 i は，i − (i と連結でない頂点) 間の 1st min の行き先を放送する ($8\ \mathrm{bit}$).
- 各頂点 i は，i − (i と連結でない頂点) 間の 2nd min の (重み, 行き先) を放送する ($56\ \mathrm{bit}$).

テクニック (4) : 初手 3 倍

定理

頂点集合を V とする.
任意の $S \subset V$ に対して, $T := V \setminus S$ とする.
$S - T$ 間の最小重みの辺は, 最小全域木に採用して良い.

1st min の辺が双方向に向いていたら, その辺を採用した後, その連結成分から出る最小重みの辺は「2nd min のうち重みの小さい方」である. この辺も採用できる！

→ 連結成分のサイズが 3 以上に！

ブルーフカ法 + 初手 3 倍 + 集計 : 9 ラウンド

0 ラウンド目 : 1st min と 2nd min を送る

1 ラウンド目以降 :

- 各頂点 i は，$\mathrm{parent}[i]$ を放送する (8 bit).
- 各頂点 i は，i − (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).

$3 \times 2^6 = 192$ なので，**7 回の放送で最小全域木を求める → 追加の 1 回の放送で，最小全域木に採用された辺の重みを放送する**

9 ラウンドでできた！

小課題 $4:3$ 倍ブルーフカ法

小課題 4 の制約「$A_{i,j}$ は $20\,\mathrm{bit}$」をうまく使えないか？

→ (最小重み, 行き先) を 2 本放送することができる！

小課題 $4 : 3$ 倍ブルーフカ法

- 各頂点 i は，$\mathrm{leader}[i]$ を放送する (8 bit).
- 各頂点 i は，i − (i と連結でない頂点) 間の 1st min と 2nd min の (重み, 行き先) を放送する ($28 + 28\ \mathrm{bit}$).
- 各連結成分ごとに，その内外を結ぶ辺の 1st min と 2nd min を求める
- 1st min の辺が双方向に向いていたら，その辺を採用した後,
  その連結成分から出る最小重みの辺は「2nd min のうち重みの小さい方」である．この辺も採用できる！

$3^5 = 243$ なので，放送 5 回 → 6 ラウンドでできた！💯

ブルーフカ法 : 9 ラウンド (再掲)

- 各頂点 i は，$\mathrm{leader}[i]$ を放送する (8 bit).
- 各頂点 i は，i − (i と連結でない頂点) 間の (最小重み, 行き先) を放送する (56 bit).
  - 自分自身へは，「採用された辺の重みの合計」を送信する (56 bit)

放送 8 回 → 9 ラウンドでできた！

どこに無駄があるか？ 🤔

ブルーフカ法 : 9 ラウンド (再掲)

連結成分のサイズ :

$$1 \to 2 \to 4 \to 8 \to 16 \to 32 \to 64 \to 128 \to 256$$

採用される辺の本数 :

$$1 \xrightarrow{128} 2 \xrightarrow{64} 4 \xrightarrow{32} 8 \xrightarrow{16} 16 \xrightarrow{8} 32 \xrightarrow{4} 64 \xrightarrow{2} 128 \xrightarrow{1} 256$$

ブルーフカ法：9 ラウンド (再掲)

連結成分のサイズ：

$$1 \to 2 \to 4 \to 8 \to 16 \to 32 \to 64 \to 128 \to 256$$

採用される辺の本数：

$$1 \xrightarrow{128} 2 \xrightarrow{64} 4 \xrightarrow{32} 8 \xrightarrow{16} \underbrace{16 \xrightarrow{8} 32 \xrightarrow{4} 64 \xrightarrow{2} 128 \xrightarrow{1} 256}_{\text{まとめてできないか？}}$$

テクニック (5) : 2 ラウンドかけて頂点を縮約

採用される辺の本数：

$$1 \xrightarrow{128} 2 \xrightarrow{64} 4 \xrightarrow{32} 8 \xrightarrow{16} \underbrace{16 \xrightarrow{8} 32 \xrightarrow{4} 64 \xrightarrow{2} 128 \xrightarrow{1} 256}_{\text{まとめてできないか？}}$$

16 頂点の連結成分が 16 個あるとき，**各連結成分間の最小重みの情報を 256 頂点で分担して放送できそう！**

ブルーフカ法 + 縮約 : 7 ラウンド

ブルーフカ法で 4 回放送し，(各連結成分のサイズ) ≥ 16 にしておく．

目標 : 連結成分 i の k 番目の頂点が，(連結成分 i) $-$ (連結成分 k) 間の最小重みを放送する．

ブルーフカ法 + 縮約 : 7 ラウンド

ブルーフカ法で 4 回放送し，(各連結成分のサイズ) ≥ 16 にしておく．

目標 : 連結成分 i の k 番目の頂点が，(連結成分 i) $-$ (連結成分 k) 間の最小重みを放送する．

↑

そのために，(連結成分 i の k 番目の頂点) は，(連結成分 k) の各頂点 j から，j $-$ (連結成分 i) 間の最小重みを受け取る．

どうやって採用された辺の重みを合計する？

ブルーフカ法 + 縮約 : 7 ラウンド

ブルーフカ法で 4 回放送し，(各連結成分のサイズ) ≥ 16 にしておく．

目標 : 連結成分 i の k 番目の頂点が，(連結成分 i) − (連結成分 k) 間の最小重みを放送する．$i = k$ **なら放送する必要はない！**

↑

そのために，(連結成分 i の k 番目の頂点) は，(連結成分 k) の各頂点 j から，j − (連結成分 i) 間の最小重みを受け取る．

ここに「採用された辺の重みの合計」を入れよう！

ブルーフカ法 + 縮約 : 7 ラウンド

ブルーフカ法で 4 回放送し，(各連結成分のサイズ) ≥ 16 にしておく．

目標 : 連結成分 i の k 番目の頂点は，$i = k$ **なら，「採用された辺の重みの合計」を放送する**

↑

そのために，(連結成分 i の k 番目の頂点) は，**いずれかの頂点から「採用された辺の重みの合計」を受け取る．**

これで辺の重みを合計できる！

ブルーフカ法 + 縮約 : 7 ラウンド

ブルーフカ法で 4 回放送し，(各連結成分のサイズ) ≥ 16 にしておく．

	$i \neq k$	$i = k$
連結成分 i の k 番目の頂点が,	(連結成分 i) − (連結成分 k) 間の最小重みを放送	「採用された辺の重みの合計」を放送
そのために, (連結成分 i の k 番目の頂点) は,	(連結成分 k) の各頂点 j から，j − (連結成分 i) 間の最小重みを受け取る	いずれかの頂点から「採用された辺の重みの合計」を受け取る

6 回放送 → 7 ラウンドでできた！

ブルーフカ法 + 縮約 : 7 ラウンド

ブルーフカ法で 4 回放送し，(各連結成分のサイズ) ≥ 16 にしておく．

	$i \neq k$	$i = k$
連結成分 i の k 番目の頂点が,	(連結成分 i) $-$ (連結成分 k) 間の最小重みを放送	「採用された辺の重みの合計」を放送
そのために, (連結成分 i の k 番目の頂点) は,	(連結成分 k) の各頂点 j から， j $-$ (連結成分 i) 間の最小重みを受け取る	いずれかの頂点から「採用された辺の重みの合計」を受け取る

6 回放送 → 7 ラウンドでできた！

本当に 6 ラウンドにできるんですか？🤔

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

実は初手 3 倍が使える！

- 各頂点 i は，i – (i と連結でない頂点) 間の 1st min の行き先を放送する ($8\ \mathrm{bit}$).
- 各頂点 i は，i – (i と連結でない頂点) 間の 2nd min の (重み, 行き先) を放送する ($56\ \mathrm{bit}$).
- 1st min の辺が双方向に向いていたら，その辺を採用した後，その連結成分から出る最小重みの辺は「2nd min のうち重みの小さい方」である．この辺も採用できる！

どうやって，採用された 1st min の辺の重みを合計する？

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

- 各頂点 i は，i − (i と連結でない頂点) 間の 1st min の行き先を放送する ($8\,\mathrm{bit}$).
- 各頂点 i は，i − (i と連結でない頂点) 間の 2nd min の (重み, 行き先) を放送する ($56\,\mathrm{bit}$).
- 1st min の辺が双方向に向いていたら，その辺を採用した後，その連結成分から出る最小重みの辺は「2nd min のうち重みの小さい方」である．この辺も採用できる！

- 採用された 1st min の辺の重みは，その辺の両端の頂点しか分からない

→ どちらかに，採用された 1st min の辺の重みの合計を保持してもらう！

- 採用された 2nd min の辺の重みは誰でも分かる

→ 頂点 0 に合計を保持してもらう！

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

- 初手 3 倍
  - 採用された 1st min の辺の重みは，その辺の一端が保持する
  - 採用された 2nd min の辺の重みは，頂点 0 が保持する
- ブルーフカ法 2 回
  - 採用された辺の重みは，頂点 0 が保持する
- **どうやって各頂点が保持している辺重みを合計する？**
- 縮約
  - **連結成分のサイズより連結成分数が多い！**

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

どうやって各頂点が保持している辺重みを合計する？

	$i \neq k$	$i = k$
連結成分 i の k 番目の頂点が,	(連結成分 i) − (連結成分 k) 間の最小重みを放送	「採用された辺の重みの合計」を放送
そのために, (連結成分 i の k 番目の頂点) は,	(連結成分 k) の各頂点 j から, j − (連結成分 i) 間の最小重みを受け取る	いずれかの頂点から「採用された辺の重みの合計」を受け取る

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

どうやって各頂点が保持している辺重みを合計する？

	$i \neq k$	$i = k$
連結成分 i の k 番目の頂点が,	(連結成分 i) — (連結成分 k) 間の最小重みを放送	「採用された辺の重みの合計」を放送
そのために, (連結成分 i の k 番目の頂点) は,	(連結成分 k) の各頂点 j から, j — (連結成分 i) 間の最小重みを受け取る	各頂点から「保持している辺重み」を受け取る

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

サイズ 12 の連結成分が最大 21 個あって，すべての情報を送れない…？

	$i \neq k$	$i = k$
連結成分 i の k 番目の頂点が，	(連結成分 i) − (連結成分 k) 間の最小重みを放送	「採用された辺の重みの合計」を放送
そのために，(連結成分 i の k 番目の頂点) は，	(連結成分 k) の各頂点 j から，j − (連結成分 i) 間の最小重みを受け取る	各頂点から「保持している辺重み」を受け取る

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

サイズ 12 の連結成分が最大 21 個あって，すべての情報を送れない…？

→ $\binom{21}{2} = 210 < 256$ だから頂点数は足りているはず！

	$i \neq k$	$i = k$
連結成分 i の k 番目の頂点が,	(連結成分 i) — (連結成分 k) 間の最小重みを放送	「採用された辺の重みの合計」を放送
そのために, (連結成分 i の k 番目の頂点) は,	(連結成分 k) の各頂点 j から，j — (連結成分 i) 間の最小重みを受け取る	各頂点から「保持している辺重み」を受け取る

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

双方向に辺を送らないように，連結成分 i は連結成分 $i \to i + \{1, 2, \ldots, 10\} \pmod N$ の辺を放送！

	$k \neq 0$	$k = 0$
連結成分 i の k 番目の頂点が，	(連結成分 i) $-$ (連結成分 $i+k$) 間の最小重みを放送	「採用された辺の重みの合計」を放送
そのために，(連結成分 i の k 番目の頂点) は，	(連結成分 $i+k$) の各頂点 j から，j $-$ (連結成分 i) 間の最小重みを受け取る	各頂点から「保持している辺重み」を受け取る

ブルーフカ法 + 縮約 + 初手 3 倍 : 6 ラウンド

- 初手 3 倍
  - 採用された 1st min の辺の重みは，その辺の一端が保持する
  - 採用された 2nd min の辺の重みは，頂点 0 が保持する
- ブルーフカ法 2 回
  - 採用された辺の重みは，頂点 0 が保持する
- 縮約 (2 ラウンド)
  - 連結成分 i は連結成分 $i \to i + \{1, 2, \ldots, 10\} \pmod N$ の辺を放送

放送 5 回 $= 6$ ラウンドでできる！ 🎉 💯

おまけ : $O(\log \log N)$ ラウンド

実はちょっとの変更で $O(\log \log N)$ ラウンドになっている.

おまけ : $O(\log \log N)$ ラウンド

実はちょっとの変更で $O(\log \log N)$ ラウンドになっている.

	$k \neq 0$	$k = 0$
連結成分 i の k 番目の頂点が,	縮約後の，連結成分 i から出る k-th min の (重み, 行き先) を放送	「採用された辺の重みの合計」を放送
そのために，(連結成分 i の k 番目の頂点) は,	各頂点 j から，j − (連結成分 i) 間の最小重みを受け取る	各頂点から「保持している辺重み」を受け取る

連結成分のサイズが x 以上なら，$(x-1)$-th min まで放送することができ，連結成分のサイズを x 倍にできる！

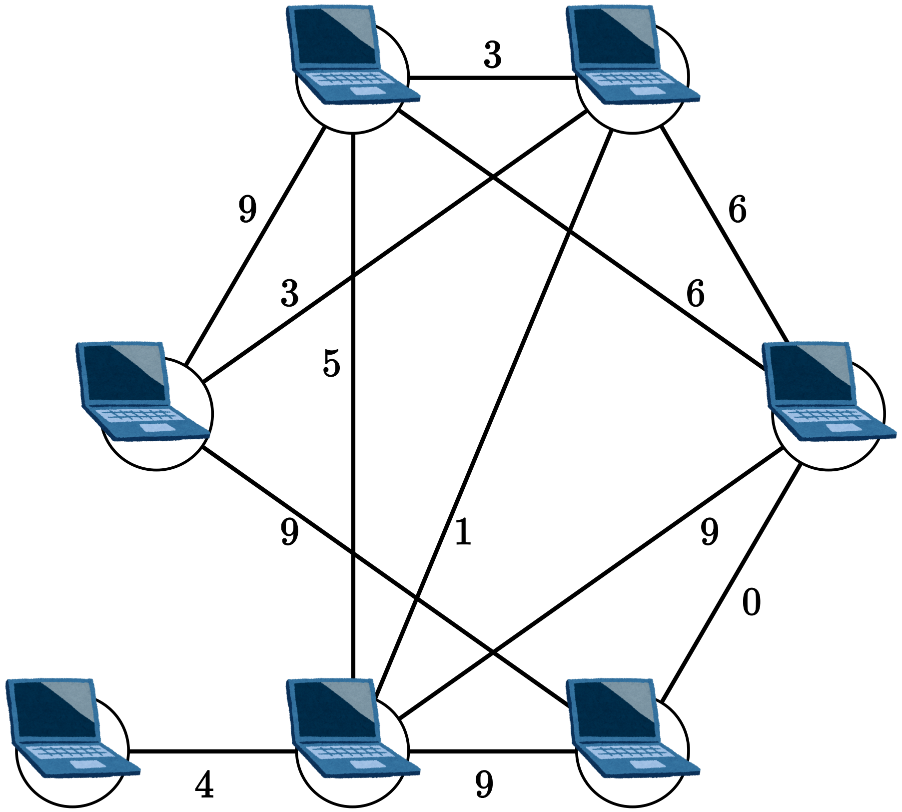
おまけ : $O(1)$ ラウンド

- 各頂点はいくらでも情報を記憶できる
- 各辺は $O(\log N)$ bit の情報を送信できる

としたとき，定数ラウンドで解けるらしい 🤔😱 [Nowicki, 2021]

おまけ : CONGEST モデル

- 入力の無向グラフ ＝ 計算ネットワーク であるようなモデル
  - 頂点 ＝ 計算機
  - 辺 ＝ 通信路
- 各頂点は各ラウンドで多項式時間の計算ができ，いくらでも記憶することができる.
- 各辺は各ラウンドで $O(\log N)$ bit の通信を行うことができる.



参考資料 : ネットワーク上の分散グラフアルゴリズムと最適化 – COSS2017

最小全域木が $O((\text{直径} + \sqrt{N}) \log N)$ ラウンドで解けるらしい [Elkin, 2020]

得点分布

得点	人数	累積
66	1	1
58	1	2
20	2	4
6	1	5
5	10	15
0	15	30