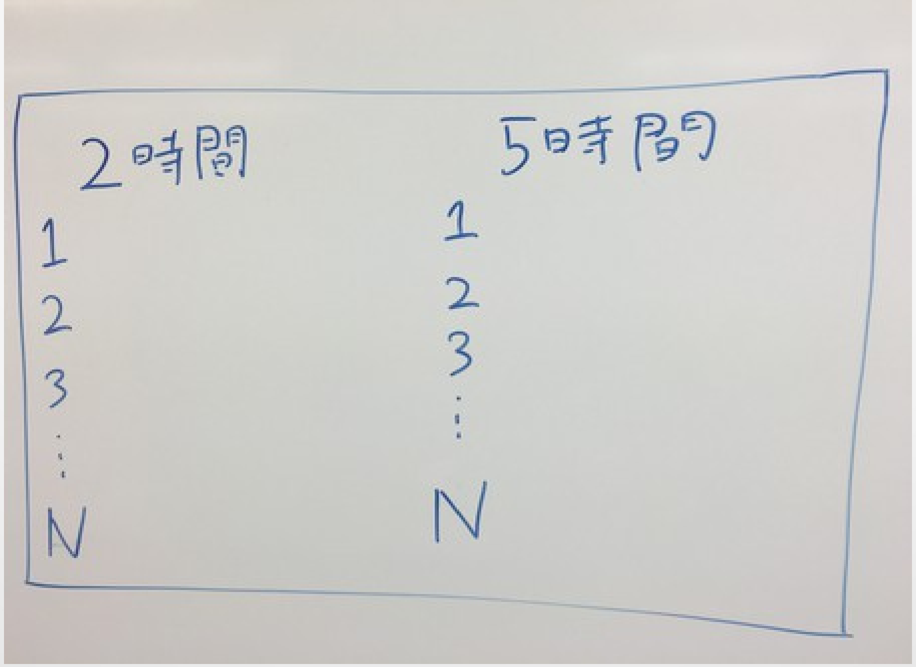
最悪の記者2

Worst Reporter 2

問題概要

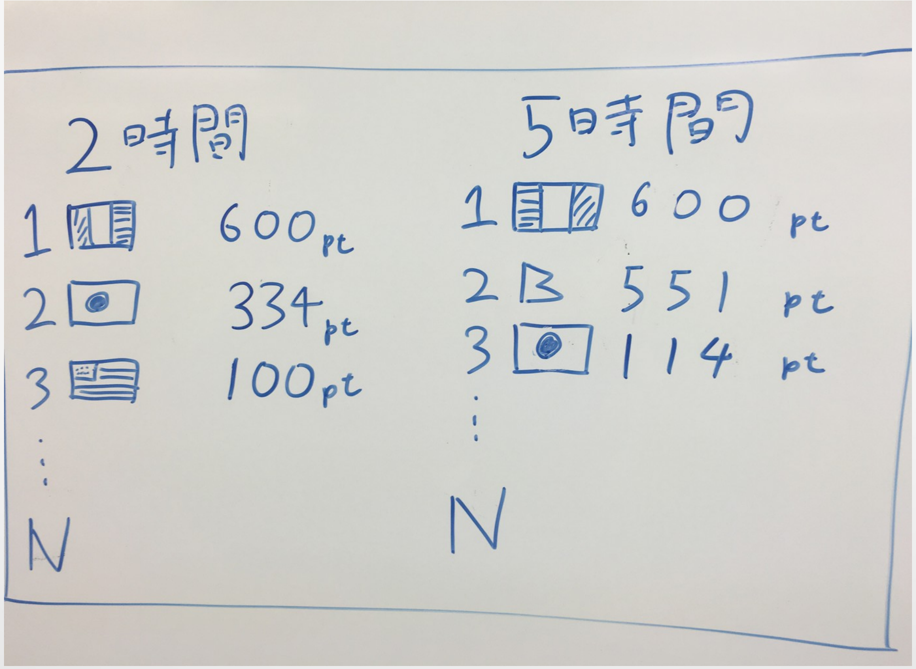
問題概要

- 競技開始 2h 後と 5h 後のN選手のランキングが与えられる



問題概要

- 競技開始 2h 後と 5h 後のN選手のランキングが与えられる
- ランキングといっても与えられるのは以下の情報のみ
  - i 位の選手の国籍と得点

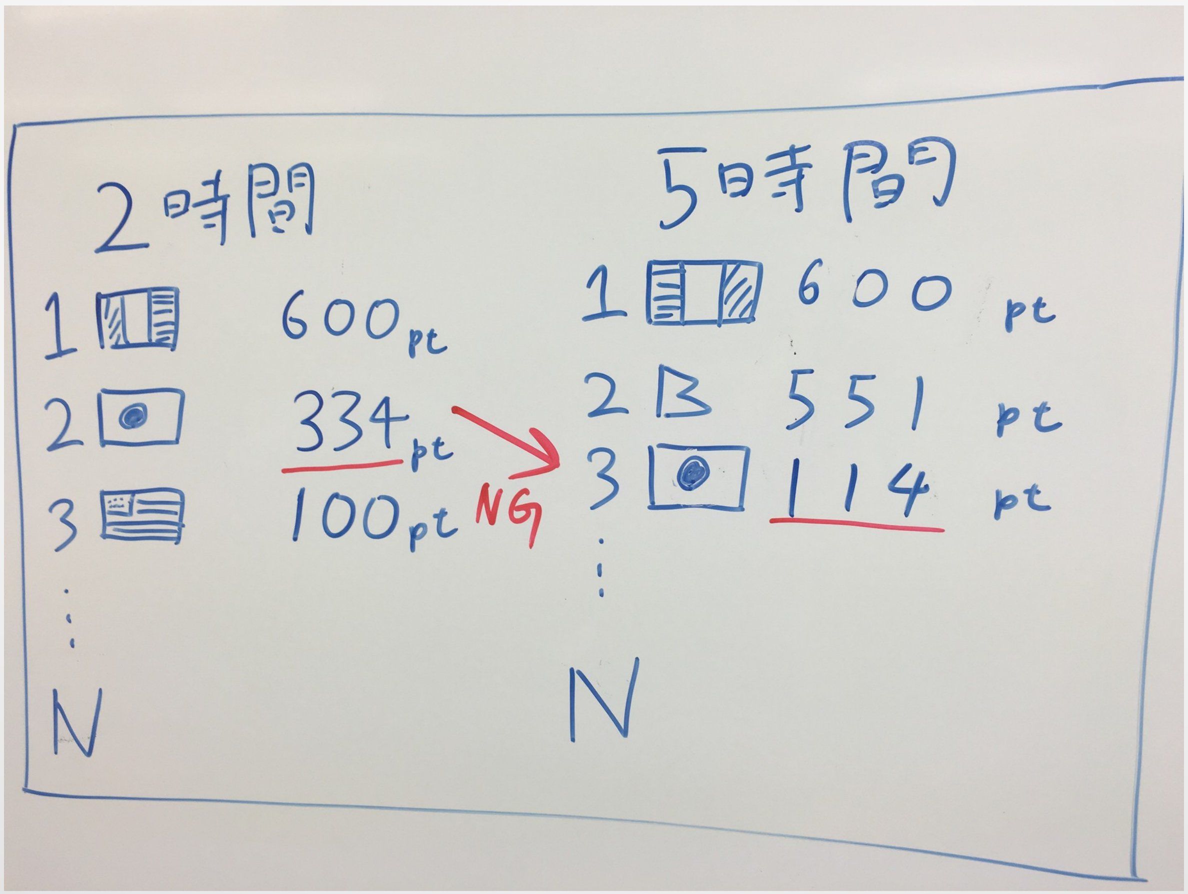


問題概要

- 競技開始 2h 後と 5h 後のN選手のランキングが与えられる
- ランキングといっても与えられるのは以下の情報のみ
  - ｉ位の選手の国籍と得点
- 同じ選手は時間経過とともに国籍が変化したり得点が減少することはない

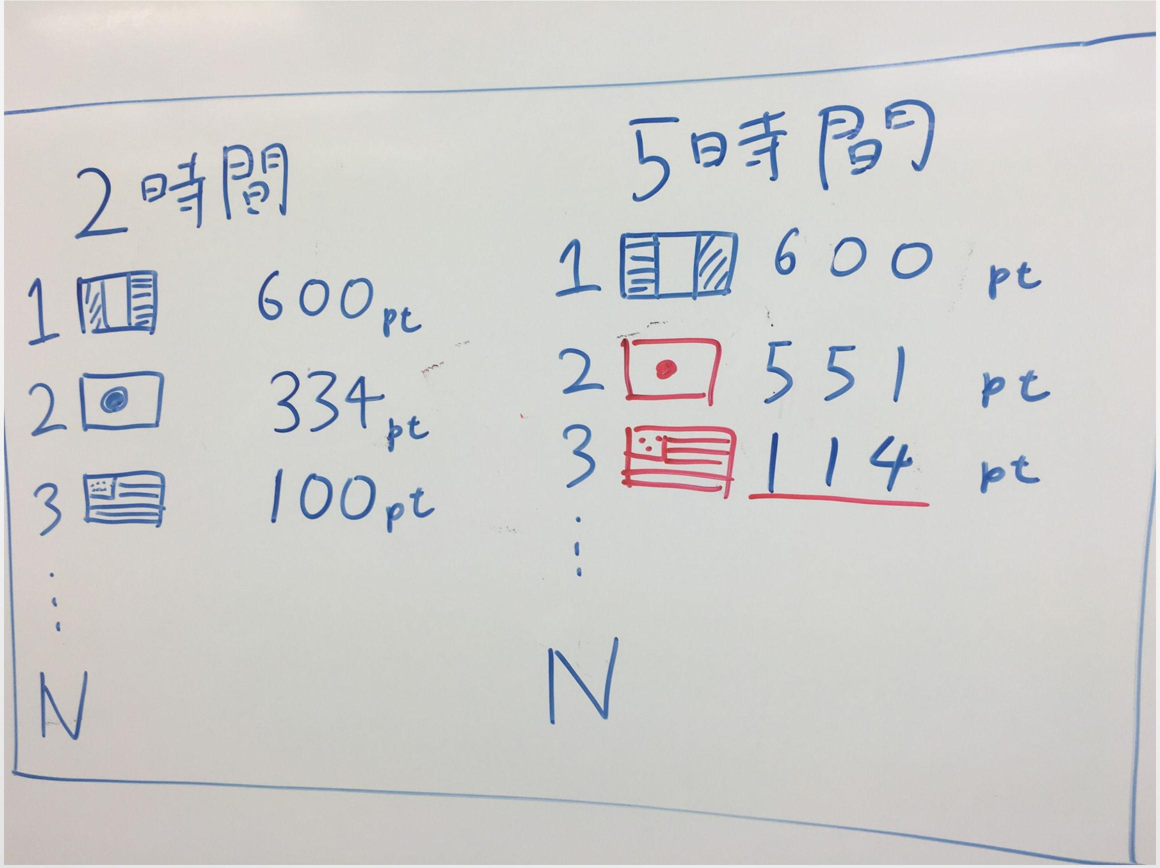
問題概要

- どうやらランキングが矛盾しているので、国籍のみを書き換えて矛盾がないランキングにしたい



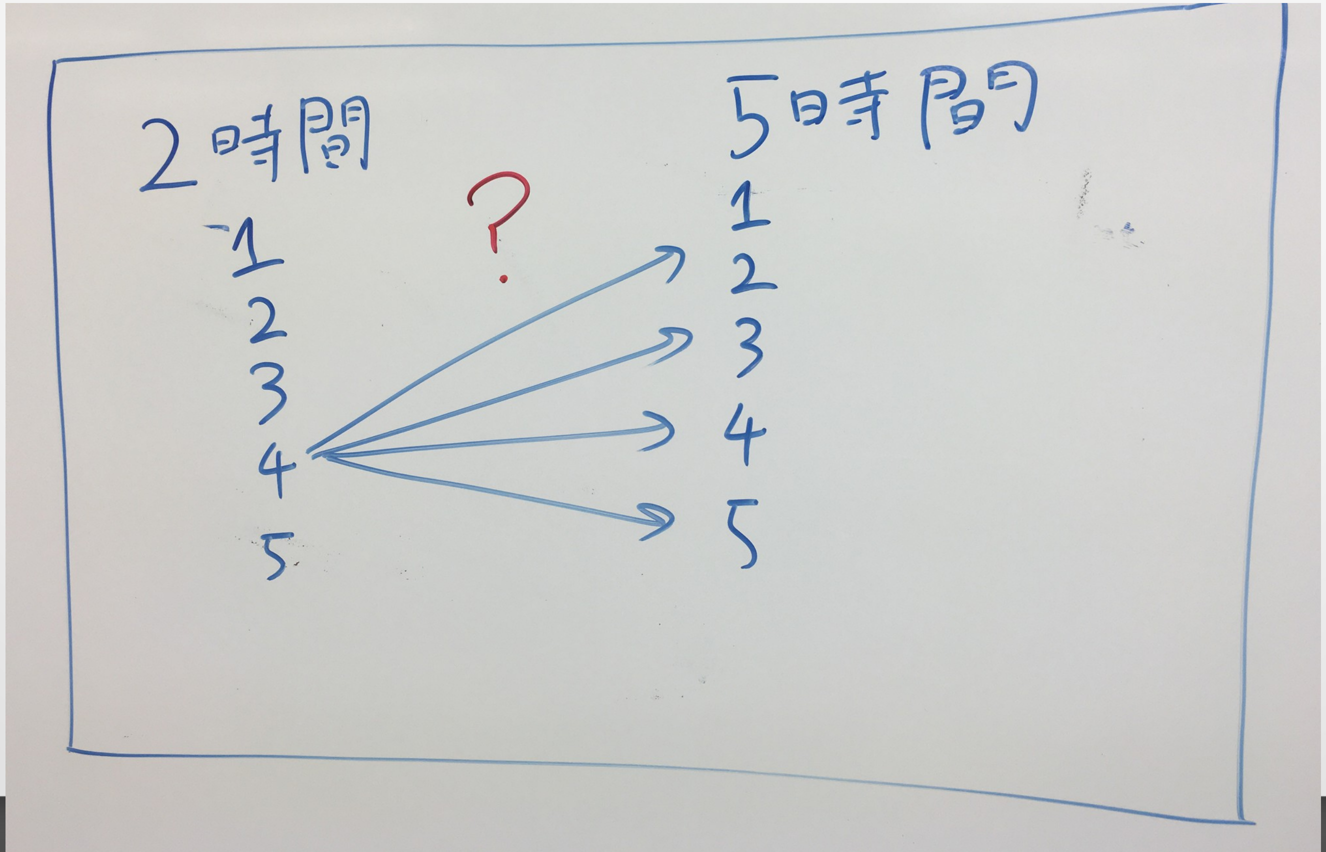
問題概要

- どうやらランキングが矛盾しているので、国籍のみを書き換えて矛盾がないランキングにしたい
- 書き換えの個数を最小化せよ



つまりどんな問題？

- 2h後のi位の選手が5h後の何位の選手に対応するかあてる問題



つまりどんな問題？

- 2h後のi位の選手が5h後の何位の選手に対応するかあてる問題
- できるだけ少ない書き換えで矛盾のない対応づけをめざす

小課題1

小課題1（N≦16）

- N ≦ 16

小課題1（N≦16）

- N ≦ 16
- 「何位の人が何位に対応するか」を全部ためせる？

小課題1（N≦16）

- N ≦ 16
- 「何位の人が何位に対応するか」を全部ためせる？
- 16! ≒ 2＊10^13

小課題1（N≦16）

- N ≦ 16
- 「何位の人が何位に対応するか」を全部ためせる？
- 16! ≒ 2 * 10^13
- おわらない

小課題1（N≦16）

- 16は階乗というよりは指数向けの数
  - 2^16 = 65536は小さめでいい数字

小課題1（N≦16）

- 16は階乗というよりは指数向けの数
  - 2^16 = 65536は小さめでいい数字

- 階乗を指数にする一般的なテクニックがある

小課題1 (N≦16)

- 16は階乗というよりは指数向けの数
  - 2^16 = 65536は小さめでいい数字

- 階乗を指数にする一般的なテクニックがある

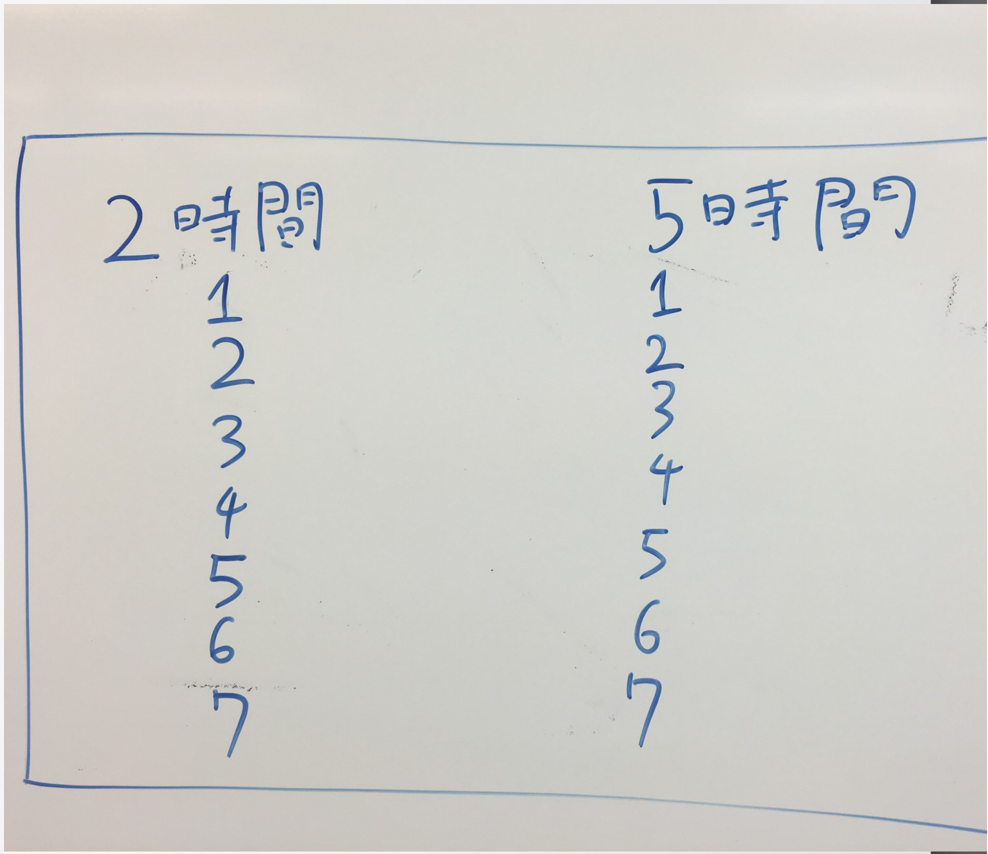
- BitDP

小課題1（N≦16）

- この問題におけるBit DPのやりかた

小課題1 (N≦16)

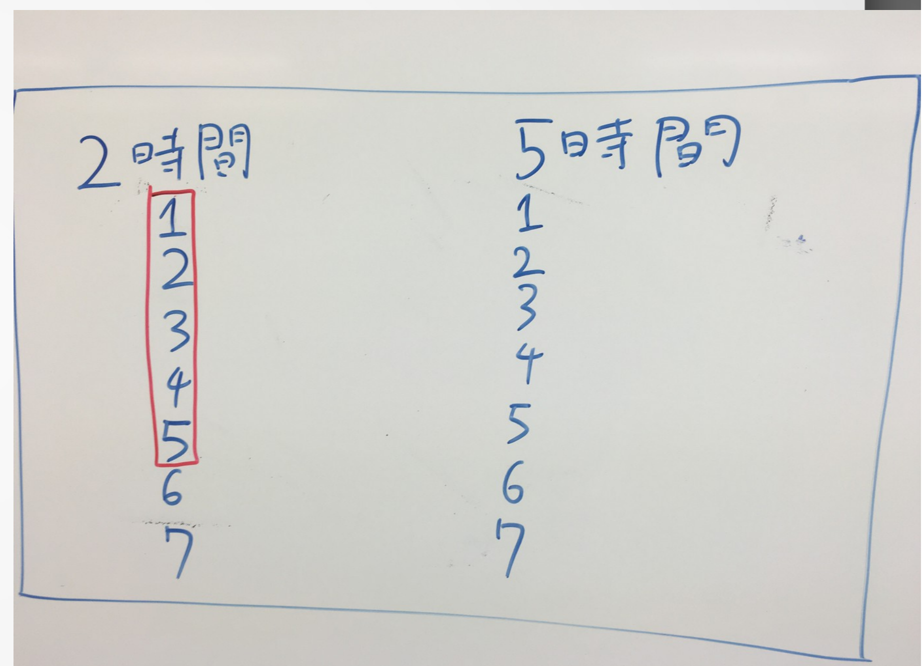
- この問題におけるBit DPのやりかた
- dp[i][N桁のbit列] =
  -
  -
  -



小課題1 (N≦16)

- この問題におけるBit DPのやりかた

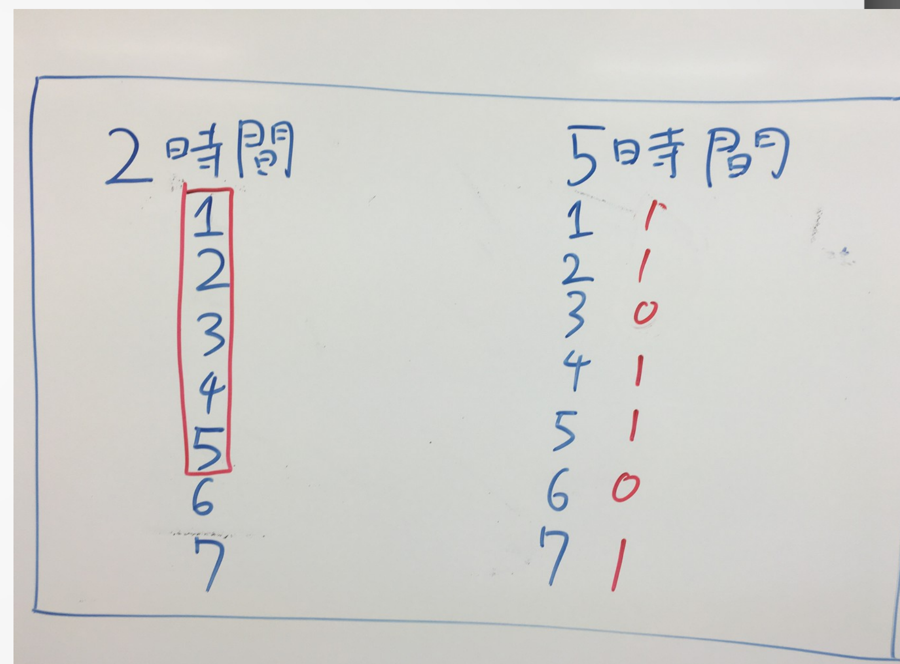
- dp[i][N桁のbit列] =
  - 2h後の1位～i位までの選手と
  -
  -



小課題1（N≦16）

- この問題におけるBit DPのやりかた

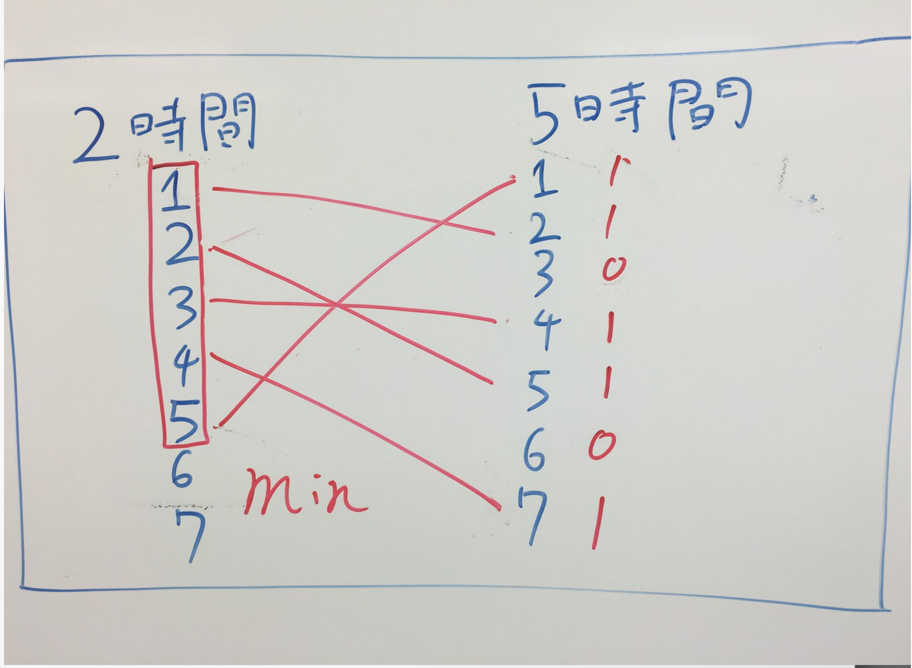
- dp[i][N桁のbit列] =
  - 2h後の1位～i位までの選手と
  - 5h後の順位でbitが立っているところにいる選手で対応付したときの
  -



小課題1（N≦16）

- この問題におけるBit DPのやりかた

- dp[i][N桁のbit列] =
  - 2h後の1位～i位までの選手と
  - 5h後の順位でbitが立っているところにいる選手で対応付したときの
  - 必要な書き換え個数の最小値



小課題1 (N≦16)

- この問題におけるBit DPの計算量

小課題1（N≦16）

- この問題におけるBit DPの計算量

- メモリはO(N*2^N)

小課題1 (N≦16)

- この問題におけるBit DPの計算量

- メモリはO(N*2^N)
- dp配列の各要素につきO(N)のループをまわす

小課題1 (N≦16)

- この問題におけるBit DPの計算量

- メモリはO(N*2^N)
- dp配列の各要素につきO(N)のループをまわす

- O(2^N * N^2)

小課題1（N≦16）

- この問題におけるBit DPの計算量

- メモリはO(N*2^N)
- dp配列の各要素につきO(N)のループをまわす

- O(2^N ＊ N^2)
  - 15点獲得！

小課題1 (N≦16)

- この問題におけるBit DPの計算量

- メモリはO(N*2^N)
- dp配列の各要素につきO(N)のループをまわす

- O(2^N ＊ N^2)
  - 15点獲得！

- ちなみにO(N ＊ N^2)でもできます

小課題2

小課題2 (N≦50)

- N ≦ 50

小課題2 (N≦50)

- N ≦ 50
  - O(N^3)とかO(N^4)くらいのアルゴリズムっぽい

小課題2 (N≦50)

- N ≦ 50
  - O(N^3)とかO(N^4)くらいのアルゴリズムっぽい

- うーん　DPかな？

小課題2 (N≦50)

- N ≦ 50
  - O(N^3)とかO(N^4)くらいのアルゴリズムっぽい
- うーん　DPかな？　実はlogとかsqrtとかが入るのかな？

小課題2 (N≦50)

- N ≦ 50
  - O(N^3)とかO(N^4)くらいのアルゴリズムっぽい
- うーん　DPかな？　実はlogとかsqrtとかが入るのかな？
- うーん わからない

小課題2（N≦50）

- この問題はどんな問題？

小課題2（N≦50）

- この問題はどんな問題？
- 2hの順位と5hの順位という「2つの」順位を「できるだけ少ない」書き換えで矛盾なく「全て」「対応付けさせる」問題

小課題2 (N≦50)

- この問題はどんな問題？
- 2hの順位と5hの順位という「2つの」順位を「できるだけ少ない」書き換えで矛盾なく「全て」「対応付けさせる」問題

- 2つの → Bipartite

小課題2（N≦50）

- この問題はどんな問題？
- 2hの順位と5hの順位という「2つの」順位を「できるだけ少ない」書き換えで矛盾なく「全て」「対応付けさせる」問題

- 2つの　→ Bipartite
- できるだけ少ない → Minimum

小課題2 (N≦50)

- この問題はどんな問題？
- 2hの順位と5hの順位という「2つの」順位を「できるだけ少ない」書き換えで矛盾なく「全て」「対応付けさせる」問題

- 2つの → Bipartite
- できるだけ少ない → Minimum
- 全て → Perfect

小課題2 (N≦50)

- この問題はどんな問題？
- 2hの順位と5hの順位という「2つの」順位を「できるだけ少ない」書き換えで矛盾なく「全て」「対応付けさせる」問題

- 2つの → Bipartite
- できるだけ少ない → Minimum
- 全て → Perfect
- 対応付け → Matching

- おや これは･･･

小課題2 (N≦50)

- この問題はどんな問題？
- 2hの順位と5hの順位という「2つの」順位を「できるだけ少ない」書き換えで矛盾なく「全て」「対応付けさせる」問題

- 2つの → Bipartite
- できるだけ少ない → Minimum
- 全て → Perfect
- 対応付け → Matching

小課題2 (N≦50)

Min Cost Perfect Matching
in
Bipartite Graph

小課題2（N≦50）

最小コスト完全2部マッチング

小課題2（N≦50）

最小コスト完全2部マッチング

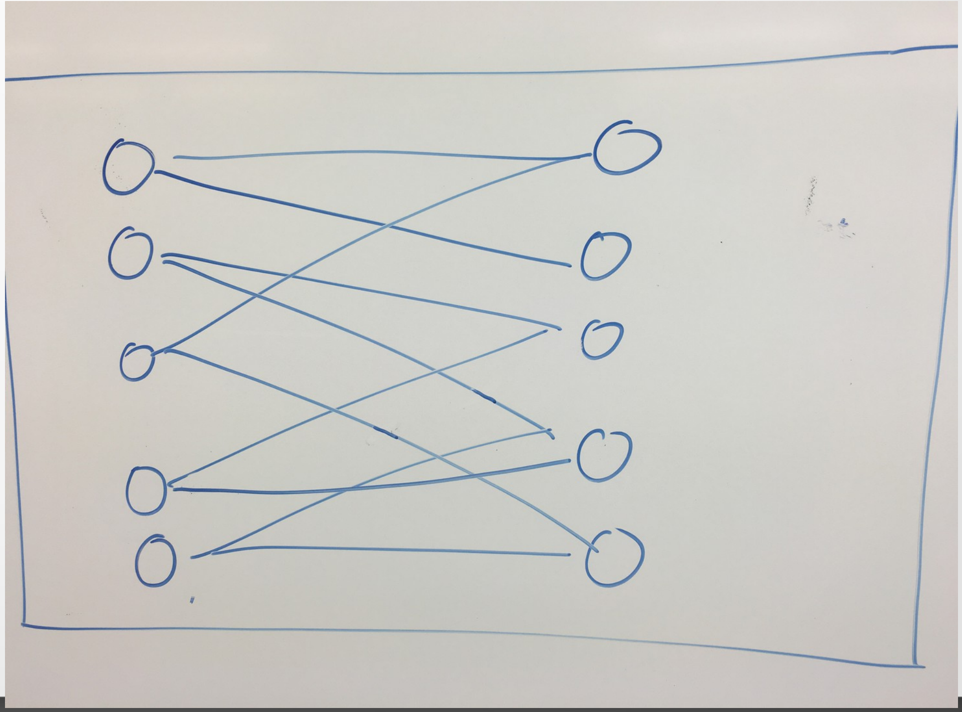
フロー
速いんだけど
微妙にバグる

小課題2 (N≦50)

- 最小コスト完全二部マッチング問題とは？

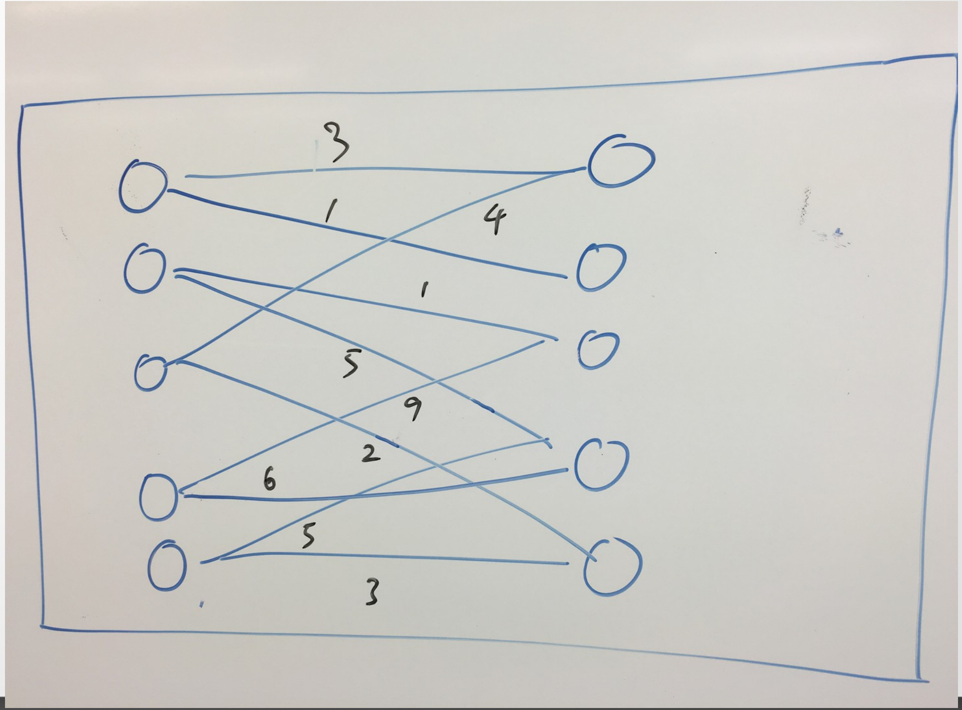
小課題2 (N≦50)

- 最小コスト完全二部マッチング問題とは？
  - 二部グラフがあります



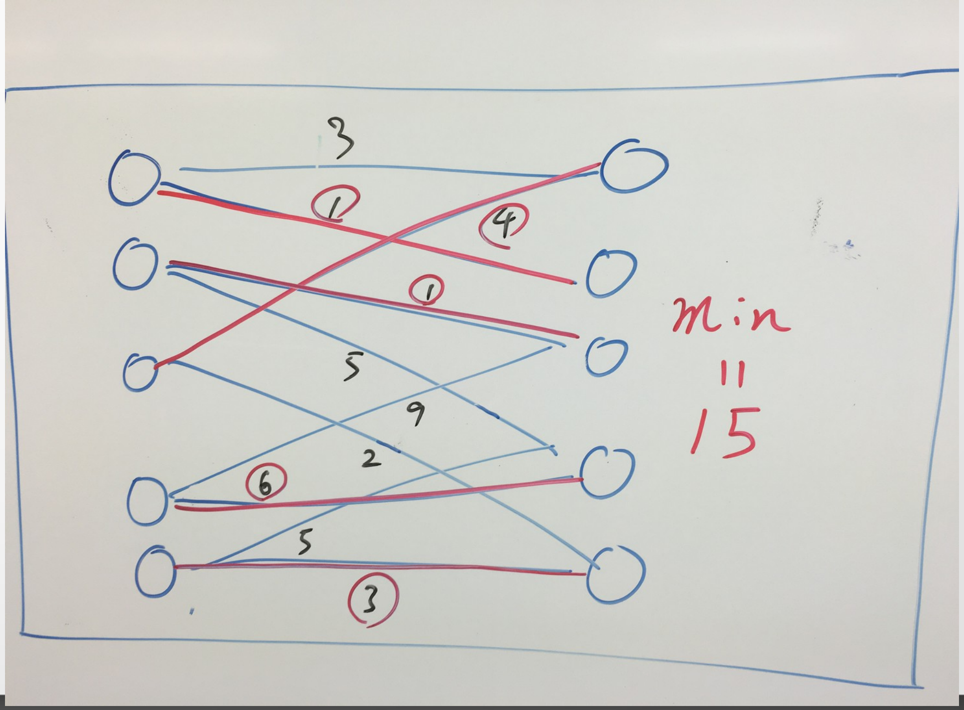
小課題2 (N≦50)

- 最小コスト完全二部マッチング問題とは？
  - 二部グラフがあります
  - 辺にはコストがあります



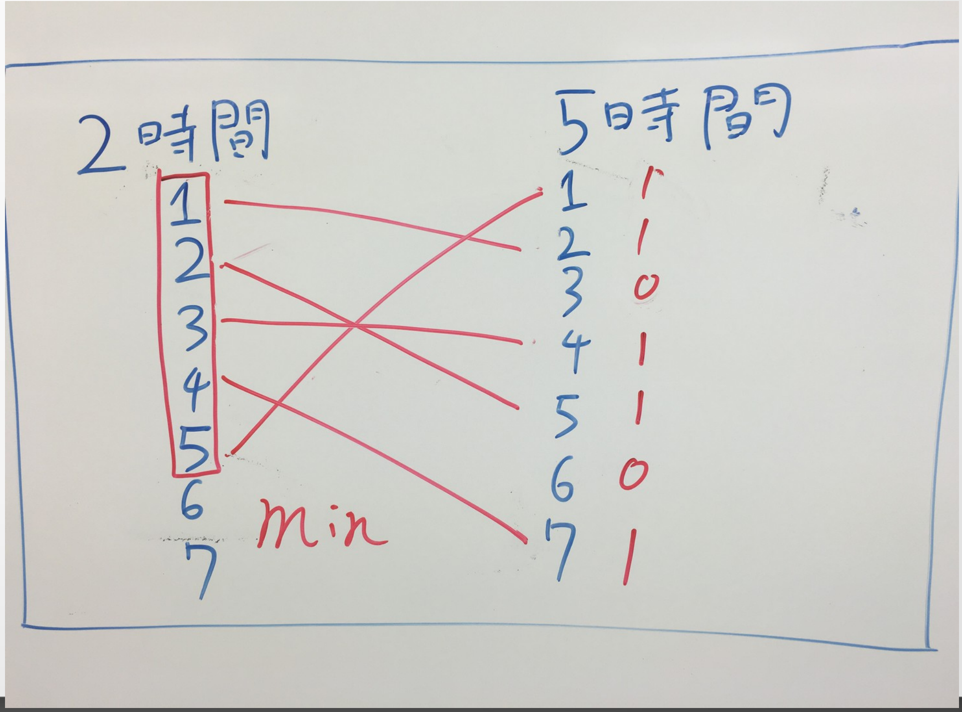
小課題2 (N≦50)

- 最小コスト完全二部マッチング問題とは？
  - 二部グラフがあります
  - 辺にはコストがあります
  - 完全マッチング(全てを一対一対応させる辺の組の選び方)のうち最小コストのものを求めてください



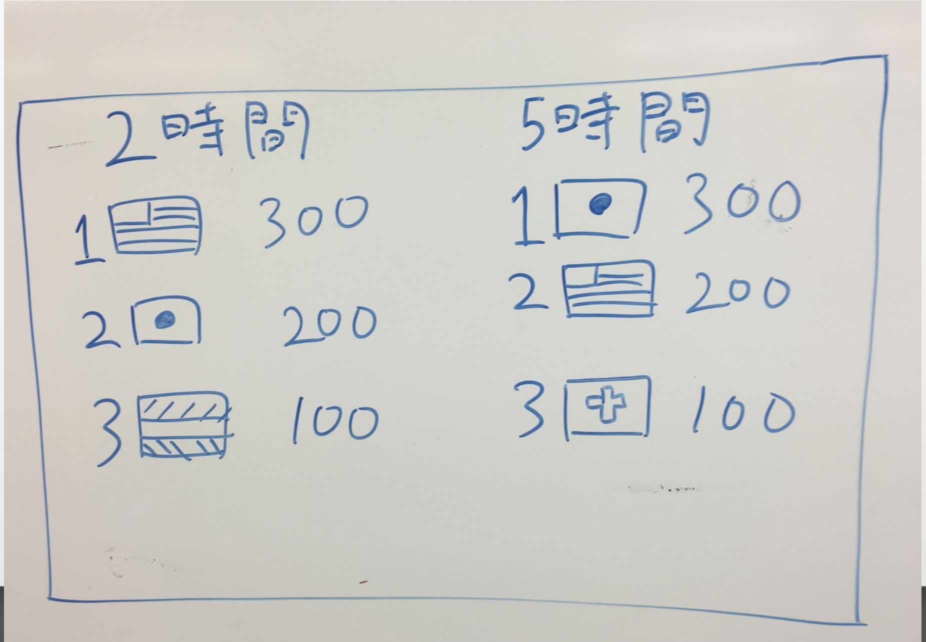
小課題2 (N≦50)

- 最小コスト完全二部マッチング問題とは？
  - 二部グラフがあります
  - 辺にはコストがあります
  - 完全マッチング(全てを一対一対応させる辺の組の選び方)のうち最小コストのものを求めてください
- 今回の問題にすごく似てる



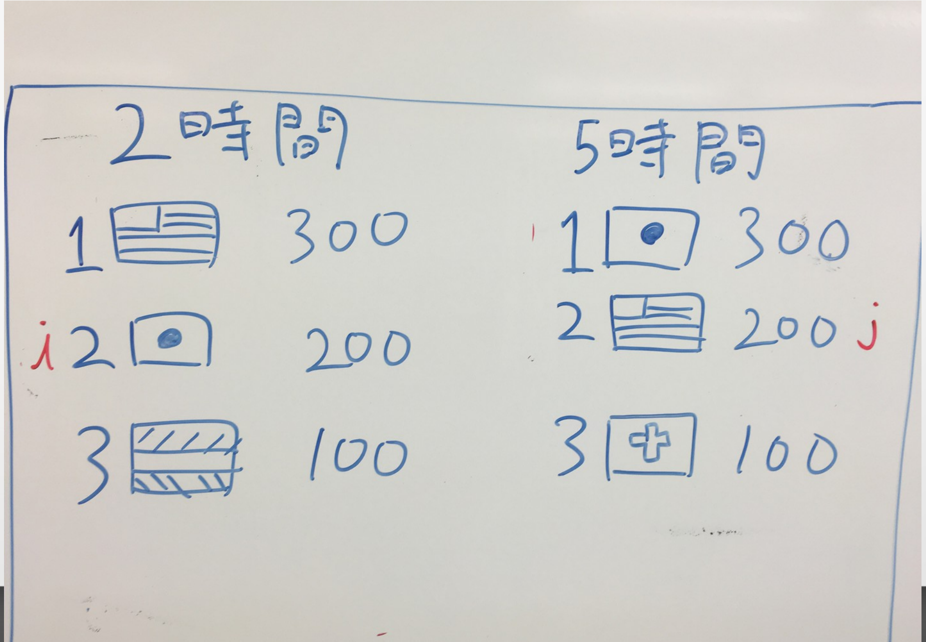
小課題2（N≦50）

- 今回の問題をどう帰着させるか



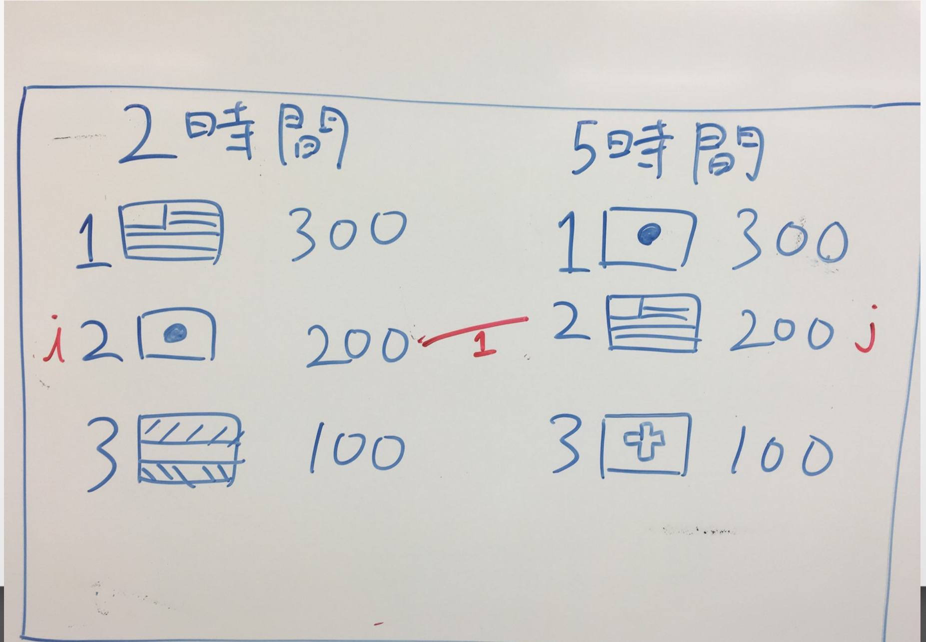
小課題2 (N≦50)

- 今回の問題をどう帰着させるか
  - 2h後のi位の選手と5h後のj位の選手を対応付けても良い時



小課題2（N≦50）

- 今回の問題をどう帰着させるか
  - 2h後のi位の選手と5h後のj位の選手を対応付けても良い時
    - 国籍が違うならコスト1の辺を張る



小課題2 (N≦50)

- 今回の問題をどう帰着させるか
  - 2h後のi位の選手と5h後のj位の選手を対応付けても良い時
    - 国籍が違うならコスト1の辺を張る
    - 国籍が同じならコスト0の辺を張る

小課題2 (N≦50)

- 今回の問題をどう帰着させるか
  - 2h後のi位の選手と5h後のj位の選手を対応付けても良い時
    - 国籍が違うならコスト1の辺を張る
    - 国籍が同じならコスト0の辺を張る
  - 対応付けられない時

小課題2 (N≦50)

- 今回の問題をどう帰着させるか
  - 2h後のi位の選手と5h後のj位の選手を対応付けても良い時
    - 国籍が違うならコスト1の辺を張る
    - 国籍が同じならコスト0の辺を張る
  - 対応付けられない時
    - 辺を張らない OR コスト無限の辺を張る

小課題2 (N≦50)

- できた2部グラフの完全マッチングは、元の問題における対応付と互いに変換可能
  - しかもマッチングのコストは、元の問題における対応付けに必要な書き換えの量と一致

小課題2（N≦50）

- できた2部グラフの完全マッチングは、元の問題における対応付と互いに変換可能
  - しかもマッチングのコストは、元の問題における対応付けに必要な書き換えの量と一致
- 帰着に成功したみたいなのであとは最小コスト完全マッチング問題を解くだけ！

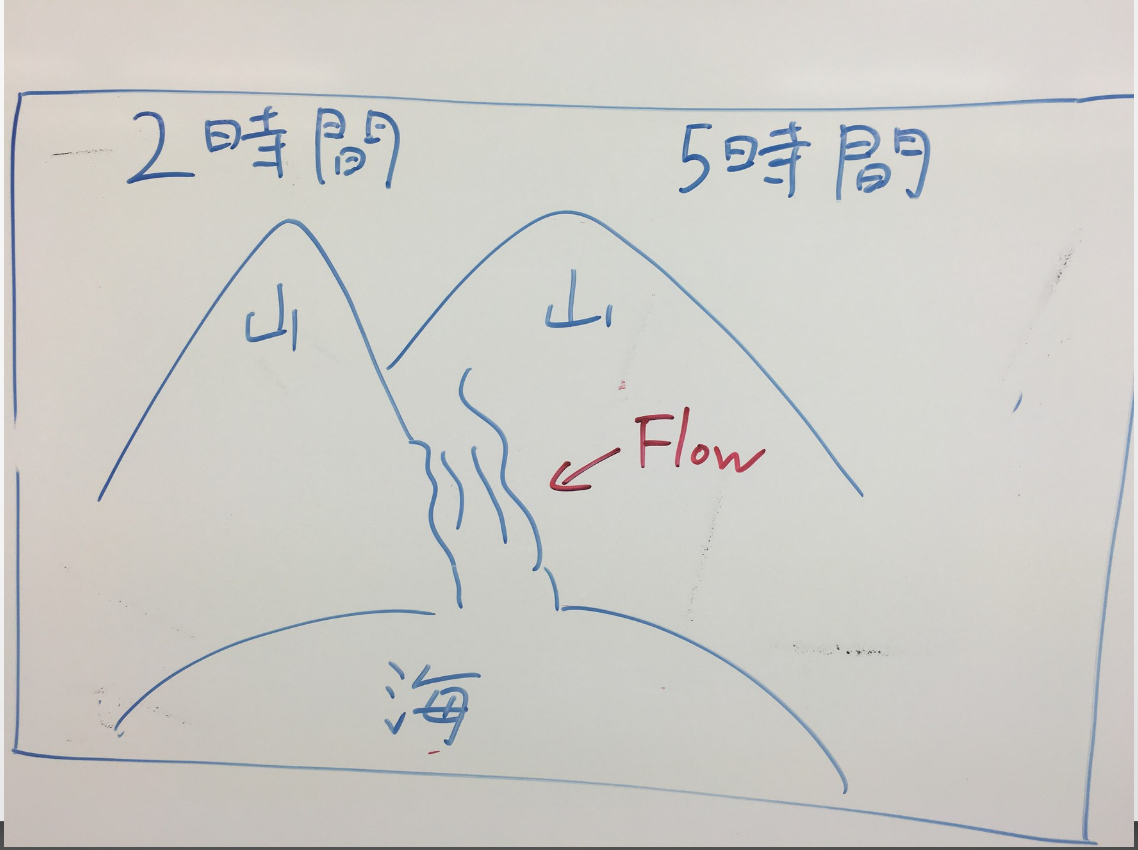
小課題2（N≦50）

- Q.最小コスト完全マッチングの求め方？

小課題2 (N≦50)

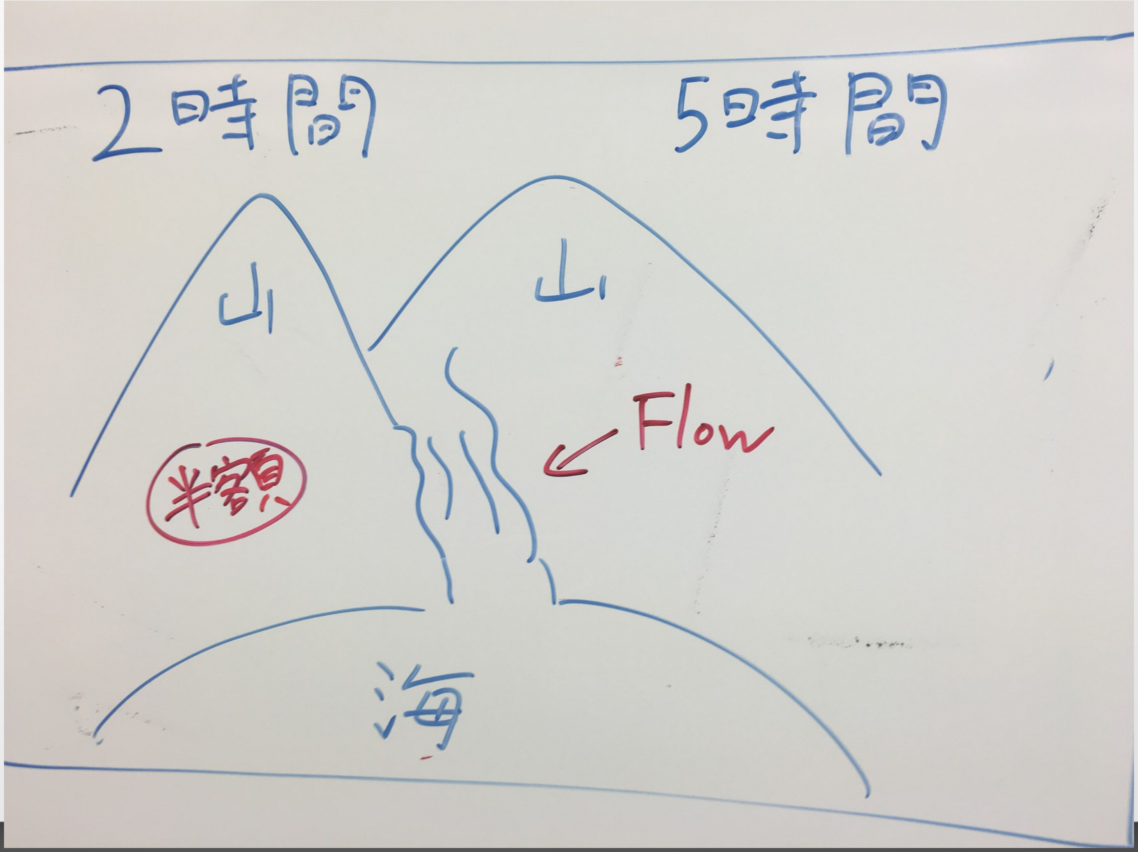
- Q.最小コスト完全マッチングの求め方？

- A.フローです。



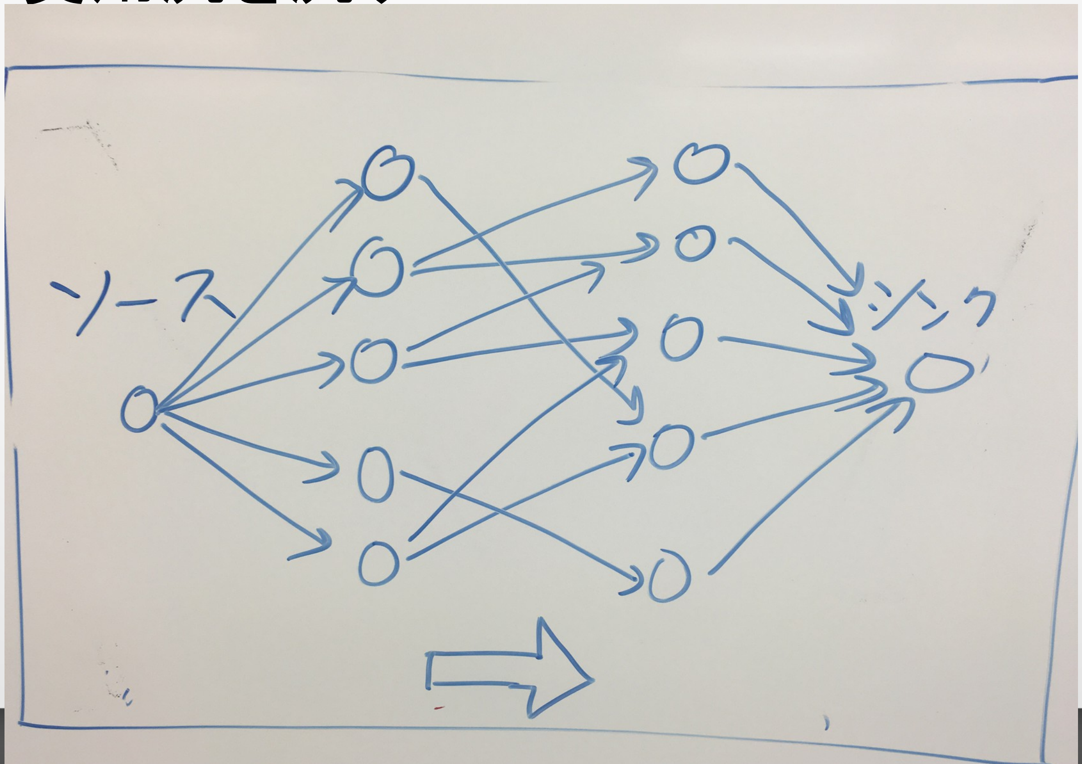
小課題2 (N≦50)

- Q.最小コスト完全マッチングの求め方？
- A.Min Cost フローです。



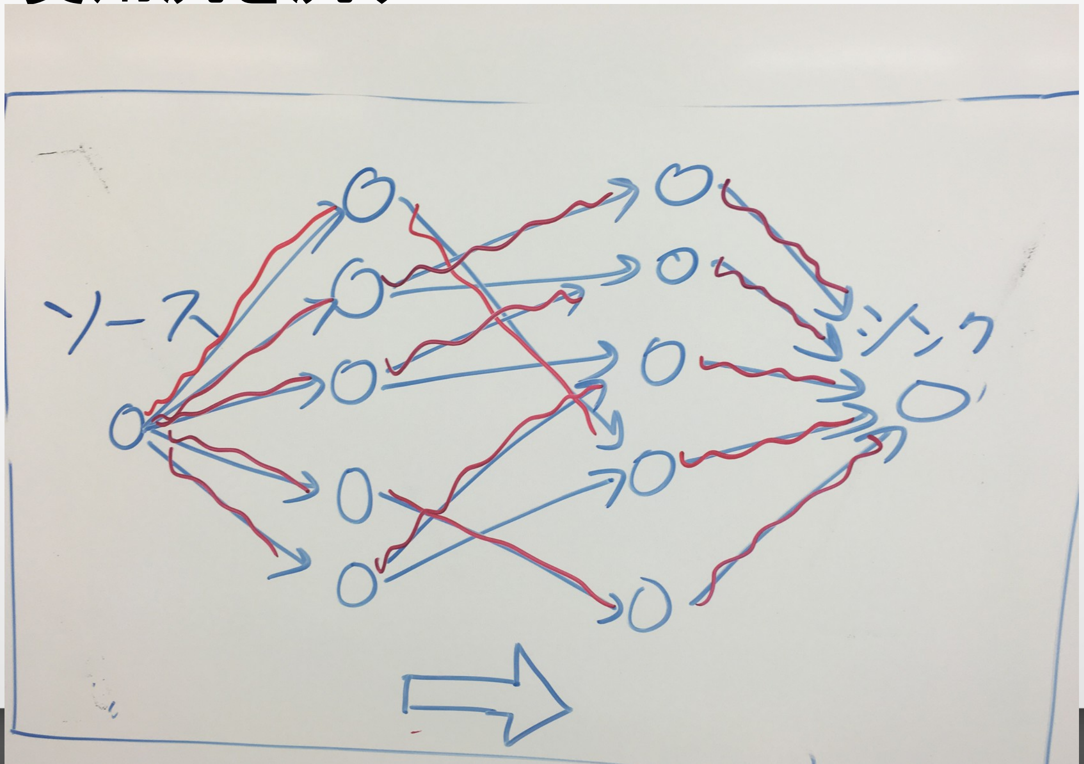
小課題2 (N≦50)

- シンクとソースを一個ずつつくって、2部グラフの各パートと辺を結ぶ
  - コストは0 流量制限は1
- シンク側からソース側に向かって、各辺に向きを付ける
- 流量Nの最小費用流を流す



小課題2 (N≦50)

- シンクとソースを一個ずつつくって、2部グラフの各パートと辺を結ぶ
  - コストは0 流量制限は1
- シンク側からソース側に向かって、各辺に向きを付ける
- 流量Nの最小費用流を流す



小課題2 (N≦50)

- シンクとソースを一個ずつつくって、2部グラフの各パートと辺を結ぶ
  - コストは0 流量制限は1
- シンク側からソース側に向かって、各辺に向きを付ける
- 流量Nの最小費用流を流す
  - 詳しいアルゴリズムは2015年のチューター講義を参照

- もちろん最小費用流パートがボトルネック
  - 計算量はO(N^3)　累計30点ゲット！

小課題3

小課題3 (N≦5000)

- N ≦ 5000

小課題3（N≦5000）

- N ≦ 5000
  - O(N^2)かな？

小課題3 (N≦5000)

- N ≦ 5000
  - O(N^2)かな？
  - フローからは離れたほうが良さそう

小課題3（N≦5000） 考察

- 書き換えるのは片側のランキングだけで良い

小課題3（N≦5000） 考察

- 書き換えるのは片側のランキングだけで良い
  - 対応付けした時のことを考えて、片側をもう片側に合わせるように書き換えを行えば良い

小課題3（N≦5000） 考察

- 書き換えるのは片側のランキングだけで良い
  - 対応付けした時のことを考えて、片側をもう片側に合わせるように書き換えを行えば良い
  - 今回は5h後のランキングのみ書き換えることにする

小課題3 (N≦5000) 考察

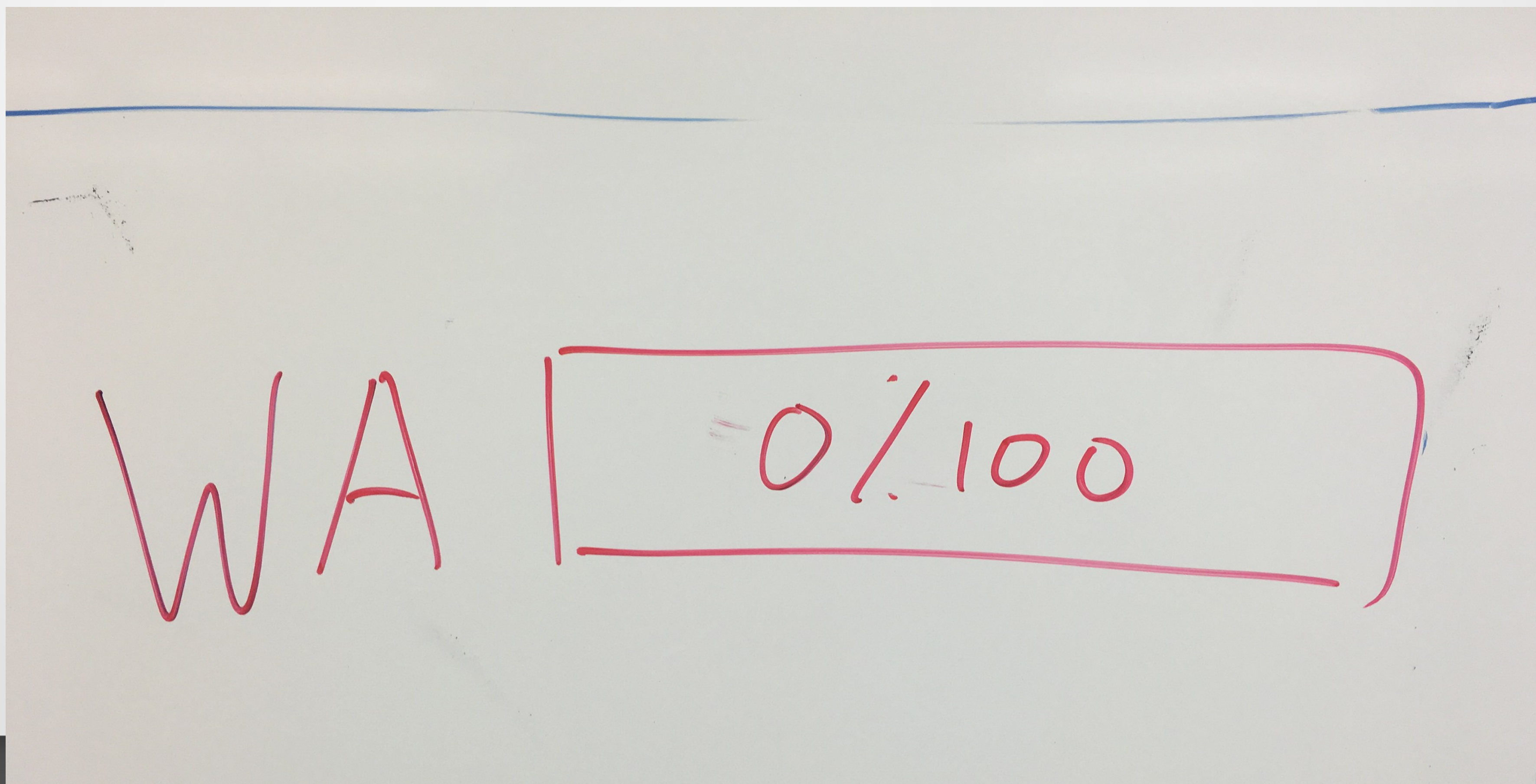
- 書き換えるのは片側のランキングだけで良い
  - 対応付けした時のことを考えて、片側をもう片側に合わせるように書き換えを行えば良い
  - 今回は5h後のランキングのみ書き換えることにする
- 5h後のランキングの中で書き換えない選手の数を最大化すれば良いことになる

小課題3（N≦5000） 考察

- 書き換えずにすむ対応付けをただひたすら多くすればよい？

小課題3（N≦5000） 考察

- 書き換えずにすむ対応付けをただひたすら多くすればよい？
  - →Wrong Answer



小課題3（N≦5000） 考察

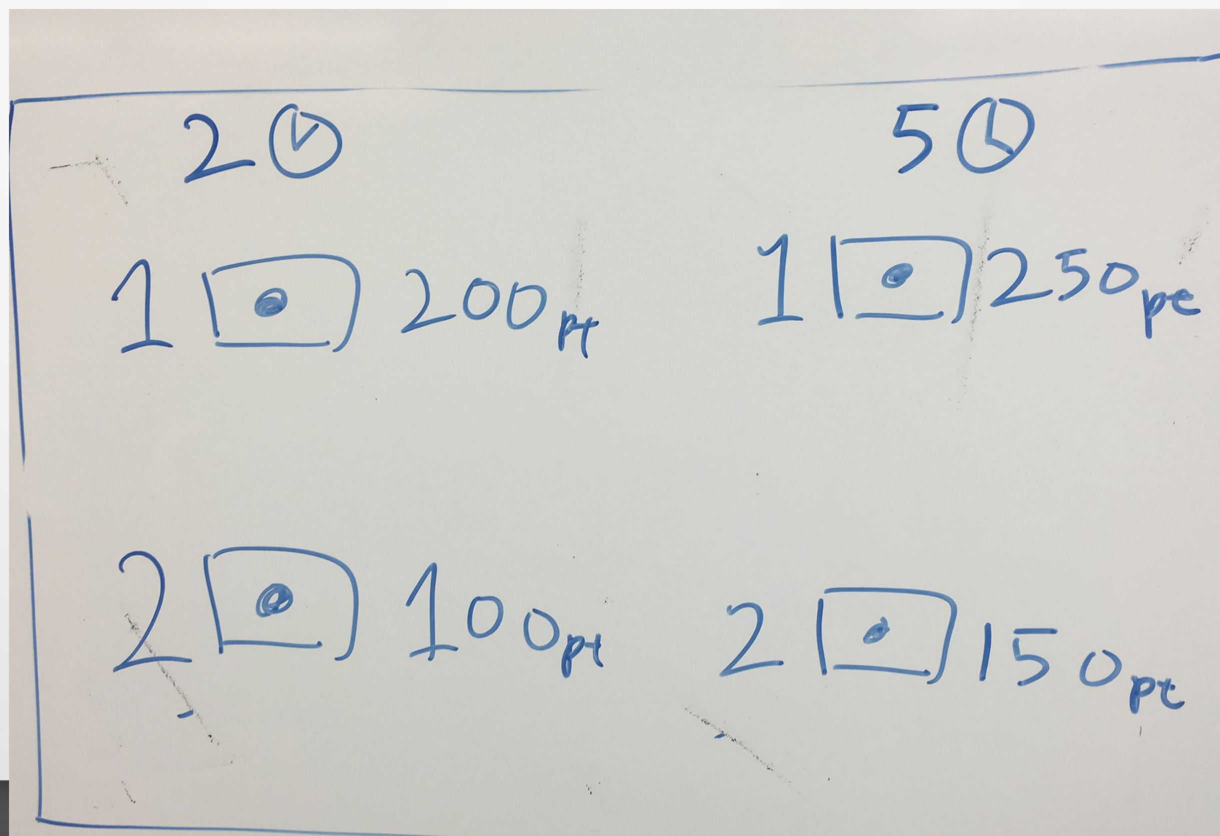
- 書き換えずにすむ対応付けをただひたすら多くすればよい？
  - →Wrong Answer
  - やたらめったら対応付けると完全マッチングが作れなくなることがある

小課題3（N≦5000） 考察

- 書き換えずにすむ対応付けをただひたすら多くすればよい？
  - →Wrong Answer
  - やたらめったら対応付けると完全マッチングが作れなくなることがある
- 例

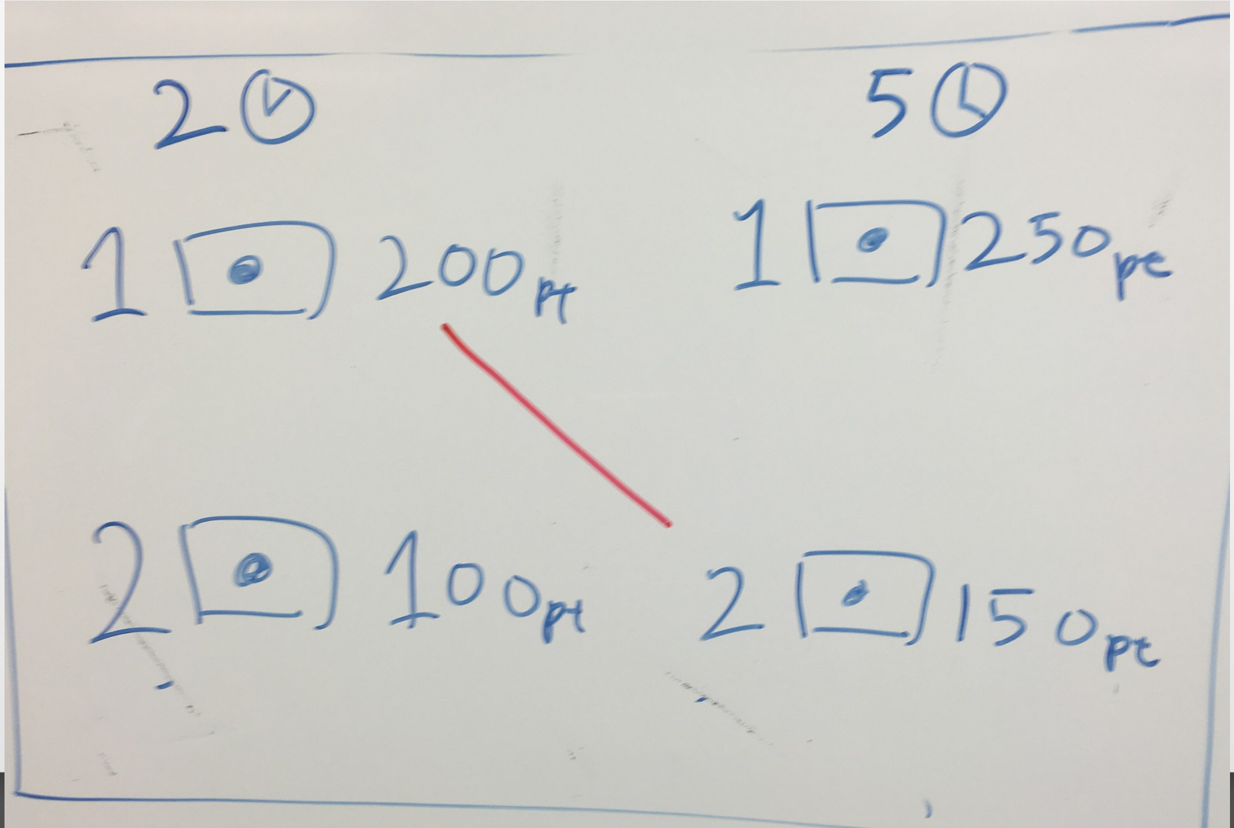
小課題3 (N≦5000） 考察

- 書き換えずにすむ対応付けをただひたすら多くすればよい？
  - →Wrong Answer
  - やたらめったら対応付けると完全マッチングが作れなくなることがある
- 例



小課題3 (N≦5000) 考察

- 書き換えずにすむ対応付けをただひたすら多くすればよい？
  - →Wrong Answer
  - やたらめったら対応付けると完全マッチングが作れなくなることがある
- 例



小課題3（N≦5000） 考察

- 書き換えずにすむ対応付けをただひたすら多くすればよい？
  - →Wrong Answer
  - やたらめったら対応付けると完全マッチングが作れなくなることがある
- 例

小課題3（N≦5000） 考察

- 完全マッチングができなくなる条件とは？

小課題3（N≦5000） 考察

- 完全マッチングができなくなる条件とは？
- あきらかに駄目なパターン
  - 両ランキングのX点以下をとりだしたとき、2h後のランキングの選手の方が少ない時

小課題3（N≦5000） 考察

- 完全マッチングができなくなる条件とは？
- あきらかに駄目なパターン
  - 両ランキングのX点以下をとりだしたとき、2h後のランキングの選手の方が少ない時
- 例

小課題3（N≦5000） 考察

- 完全マッチングができなくなる条件とは？
- あきらかに駄目なパターン
  - 両ランキングのX点以下をとりだしたとき、2h後のランキングの選手の方が少ない時
- 実は全てのXでこれが成り立っていなければ完全マッチングを作ることができる。
  - 2h後から5h後にかけて誰も順位が変わらないように対応付ければ良い

小課題3（N≦5000） 考察

- 完全マッチングができるように気をつけながら、書き換えずにすむ対応付けをひたすら多く結んでいけば良い？

小課題3（N≦5000） 考察

- 完全マッチングができるように気をつけながら、書き換えずにすむ対応付けをひたすら多く結んでいけば良い？
  - WrongAnswer

小課題3（N≦5000） 考察

- 完全マッチングができるように気をつけながら、書き換えずにすむ対応付けをひたすら多く結んでいけば良い？
  - WrongAnswer
- 反例

小課題3（N≦5000） 考察

- 完全マッチングができるように気をつけながら、書き換えずにすむ対応付けをひたすら多く結んでいけば良い？
  - WrongAnswer
- 反例

小課題3（N≦5000） 考察

- 順番を気にすればOKかもしれない

小課題3（N≦5000） 考察

- 順番を気にすればOKかもしれない
- 順番として考えられるものが得点ぐらいしか思いつかない

小課題3（N≦5000） 考察

- 順番を気にすればOKかもしれない
- 順番として考えられるものが得点ぐらいしか思いつかない

ここでいきなり解法を思いつきます

小課題3（N≦5000） 解法

- 5h後のランキングについて点数が低い選手から
  - 書き換えずに済むなら書き換えない
    - このとき、できるだけ点が高い相手と対応付ける

  - 書き換えなければならないなら書き換える
    - 何に書き換えるかは考えなくて良い

というふうに貪欲に決めていくと答えが求まる。

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3（N≦5000） 図

- 実際に以下の例でやってみる

小課題3 (N≦5000) アルゴリズム

- もう一度アルゴリズムをまとめると

- 点数が低い選手から順番に
  - 書き換えずに対応付けできるなら、できるだけ点数が高い選手と対応付ける
  - 書き換えなければならないなら放置

小課題3 (N≦5000) アルゴリズム

- もう一度アルゴリズムをまとめると

- 点数が低い選手から順番に
  - 書き換えずに対応付けできるなら、できるだけ点数が高い選手と対応付ける
  - 書き換えなければならないなら放置

小課題3（N≦5000） アルゴリズム

- もう一度アルゴリズムをまとめると
- 点数が低い選手から順番に
  - 書き換えずに対応付けできるなら、できるだけ点数が高い選手と対応付ける
  - 書き換えなければならないなら放置
- 貪欲法だが、この赤字の部分がすこし怪しい（非自明）

小課題3（N≦5000） 貪欲法のミソ

- 貪欲法は証明が難しいことが多い

小課題3（N≦5000） 貪欲法のミソ

- 貪欲法は証明が難しいことが多い
- 貪欲法を証明するときによく使えるキーワードがある

小課題3（N≦5000） 貪欲法のミソ

- 貪欲法は証明が難しいことが多い
- 貪欲法を証明するときによく使えるキーワードがある

「交換しても悪化しない」

小課題3（N≦5000） 証明

- 書き換えずに対応付けできるなら、できるだけ点数が高い選手と対応付ける

小課題3（N≦5000） 証明

- 書き換えずに対応付けできるなら、できるだけ点数が高い選手と対応付ける

小課題3（N≦5000） 証明

- 書き換えずに対応付けできるなら、できるだけ点数が高い選手と対応付ける
  - 点数が低い人と対応付けした最適解があるとする
  - より点数が高い人と交換しても悪化しない

小課題3（N≦5000） 証明

- 書き換えずに対応付けできるなら、できるだけ点数が高い選手と対応付ける
  - あえて書き換えて対応付けるような最適解があるとする
  - 1人でも書き換えずに対応付け可能な選手がいるなら、その選手と交換しても悪化しない

小課題3 (N≦5000) 実装

- 最終的に答えが格納される変数をAnsとする

- Ans = N
- 点数が低い順に対応付けを決めていく
  - できるだけ点数が高い同じ国籍の選手と対応付けできるか確認する
  - できたらその選手と対応付けて Ansを1減らす
  - できなかったら放置

小課題3 (N≦5000) 実装

- 最終的に答えが格納される変数をAnsとする

- Ans = N
- 点数が低い順に対応付けを決めていく
  - できるだけ点数が高い同じ国籍の選手と対応付けできるか確認する
  - できたらその選手と対応付けて Ansを1減らす
  - できなかったら放置

小課題3（N≦5000） 実装

- 対応付け可能かどうかの確認方法
  - まだ対応付けが完了していない選手だけをみて、先ほど示した「完全マッチングがでる条件」を満たしているか確認

全てのXについて以下が成り立つ

両ランキングのX点以下をとりだしたとき、2h後のランキングの選手の方が多い

  - 各XについてX点以下の選手の人数を両ランキングで求めれば良い
  - これはO(N)で簡単に実装できる

小課題3 (N≦5000) 実装

- 確認部分がボトルネックとなり全体の計算量は
  - O(N^2) 累計60点獲得！

小課題4 (N≦20000)

- 先ほどO(N)で実装していた部分
  - 同じ国籍、適切な点数の中で最も点が高い選手を見つける
  - 「完全マッチングが出来る条件」の判定
- これらをO(logN)とかO(1)にできれば解決

小課題4 (N≦20000)

- 同じ国籍、適切な点数の中で最も点が高い選手を見つける
  - 国籍ごとに、まだだれともマッチングされてない選手の2h後の得点を適切なデータ構造で管理すれば良い
  - やりたい操作は、値の追加、最大値の検索、最大値の削除
  - setやpriority_queueを使えばO(logN)で実装できる

  - 点数が小さいほうから見ている関係で、実は追加された値はその時点で最大値になる
  - StackでO(1)で実装できる

小課題4 (N≦20000)

- 「完全マッチングができる条件」の判定
  - 小課題3での実装は
  - 各Xについて X点以下の人数を両ランキングで求める

全てのXについて以下が成り立つ

両ランキングのX点以下をとりだしたとき、2h後のランキングの選手の方が多い

小課題4（N≦20000）

- 「完全マッチングができる条件」の判定
  - 小課題3での実装は「各XについてX点以下の人数を両ランキングで求める」というもの
  - 実際には大小関係しか気にしないので、2h後の人数ー5h後の人数だけを管理すればよい
    - 符号を見れば大小関係がわかる
  - 負の値があるかどうかは最小値の符号を見れば確認できる
  - この数列はどのように更新されていくのだろうか？

小課題4 (N≦20000)

- 「完全マッチングができる条件」の判定
  - 小課題3での実装は「各XについてX点以下の人数を両ランキングで求める」というもの
  - 実際には大小関係しか気にしないので、2h後の人数ー5h後の人数だけを管理すればよい
    - 符号を見れば大小関係がわかる
  - 負の値があるかどうかは最小値の符号を見れば確認できる
  - この数列はどのように更新されていくのだろうか？

小課題4（N≦20000）

- X点以下の2hでの人数−5hでの人数
  - 2hで30点、5hで51点の選手が対応付けられたとする

小課題4（N≦20000）

- X点以下の2hでの人数−5hでの人数
  - 2hで30点、5hで51点の選手が対応付けられたとする

小課題4（N≦20000）

- X点以下の2hでの人数-5hでの人数
  - 2hで30点、5hで51点の選手が対応付けられたとする
  - 実際に計算すると以下のように値が変わる

小課題4 (N≦20000)

- X点以下の2hでの人数-5hでの人数
  - 2hで30点、5hで51点の選手が対応付けられたとする
  - 実際に計算すると以下のように値が変わる
  - おっこれは(Day1の記憶が想起される

小課題4 (N≦20000)

Segment Tree

小課題4 (N≦20000)

- 2hでA点の選手、5hでB点の選手を対応付けられるか確認したい
- 先ほどの差の配列を観てA以上B未満の範囲の最小値が0より大きいか確認する
  - 大きいならその区間の値を1減らして、対応付けて良い
  - 最小値が0なら対応付けしてはならない
    - 完全マッチングが崩れる

小課題4 (N≦20000)

- 2hでA点の選手、5hでB点の選手を対応付けられるか確認したい
- 先ほどの差の配列を観てA以上B未満の範囲の最小値が0より大きいか確認する
  - 大きいならその区間の値を1減らして、対応付けて良い
  - 最小値が0なら対応付けしてはならない
    - 完全マッチングが崩れる

小課題4（N≦20000） 解法まとめ

- Ans = N
- 5hでの点数が低い順に選手を見ていく
  - 国籍を書き変えずに対応付けができるかを確認する
    - このときに先ほどのSegment Treeを使う
  - もし対応付けできるなら、なるべく高い人と対応付けする
    - ここはSet等を使う
    - Ans -= 1
  - そうでないなら対応付けしない
- Ansが答えになっている

小課題4（N≦20000） 解法まとめ

- Ans = N
- 5hでの点数が低い順に選手を見ていく
  - 国籍を書き変えずに対応付けができるかを確認する
    - このときに先ほどのSegment Treeを使う
  - もし対応付けできるなら、なるべく高い人と対応付けする
    - ここはSet等を使う
    - Ans -= 1
  - そうでないなら対応付けしない
- Ansが答えになっている
- 総計算量はO(NlogN)
  - 満点獲得！

得点分布