第３問 「選挙で勝とう」

解説：米田 優峻（E869120）

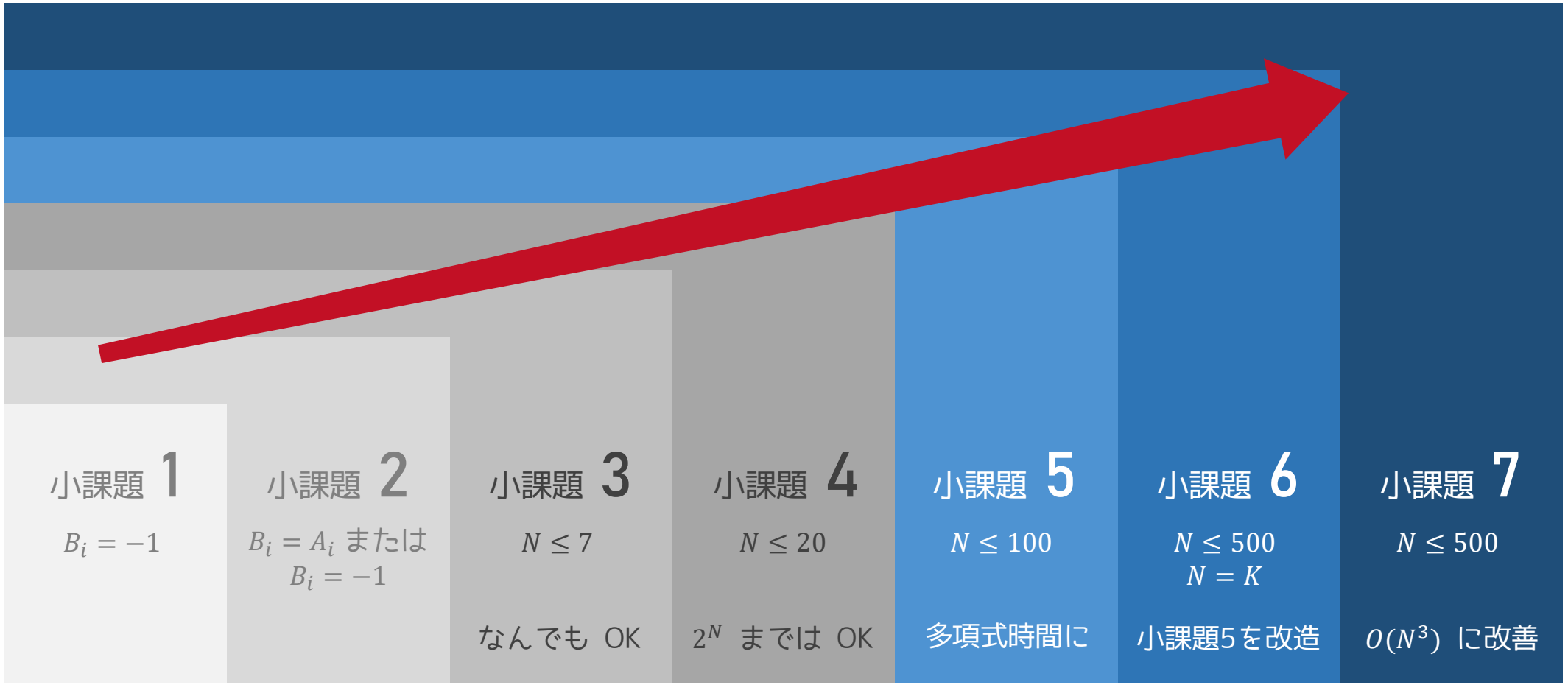
0 問題概要

- N 個の州で選挙活動をします
  - 州 i では、合計演説時間が A_i 時間に達すると票が得られます
  - 州 i では、合計演説時間が B_i 時間に達すると協力者が得られます

- 合計 K 票得るために、最小で何時間必要ですか？


演説をする理恵さん

0 満点へのロードマップ



小課題 1

$$B_i = -1$$

1 小課題 1 の解法

- $B_i = -1$ なので、協力者がいない
  - したがって、全部一人で演説する必要がある

- 票を得るのにかかる時間が短い州から取っていくと良い

1 小課題 1 の解法

- $B_i = -1$ なので、協力者がいない
  - したがって、全部一人で演説する必要がある

- 票を得るのにかかる時間が短い州から取っていくと良い

州 1	州 2	州 3	州 4	州 5	州 6	州 7
4時間	11時間	6時間	12時間	36時間	11時間	20時間

たとえば左の例で $K = 4$ 票集めたい場合、票を取るのが簡単な州 $1, 2, 3, 6$ を取れば良い

かかる時間は $4 + 11 + 6 + 11 = 32$ 時間

1 小課題 1 の実装


```cpp
#include <iostream>
using namespace std;

int N, K;
int A[509], B[509];

int main() {
    // 入力
    cin >> N >> K;
    for (int i = 0; i < N; i++) {
        cin >> A[i] >> B[i];
    }

    // ソート
    sort(A, A + N);

    // 答えを出力
    int Answer = 0;
    for (int i = 0; i < K; i++) {
        Answer += A[i];
    }
    cout << Answer << endl;
    return 0;
}
```

- 左のように、ソートを使うと簡単に実装することができます
- 皆さん、JOI では取れる部分点を全部取りましょう。

5点

小課題 2

$B_i = -1$ または $B_i = A_i$

2 小課題 2 の解法

- $B_i = -1$ または $B_i = A_i$
  - 協力者がいない(タイプ A)、または票と協力者が同時に得られる(タイプ B)

- このようなケースでは、次の 3 つのことがいえる:
  - タイプ A の中では、簡単な州 [A_i が小さい州] から取った方が良い

2 小課題 2 の解法

- $B_i = -1$ または $B_i = A_i$
  - 協力者がいない(タイプ A)、または票と協力者が同時に得られる(タイプ B)

- このようなケースでは、次の 3 つのことがいえる:
  - タイプ A の中では、簡単な州 [A_i が小さい州] から取った方が良い
  - タイプ B の中では、簡単な州 [A_i が小さい州] から取った方が良い

2 小課題 2 の解法

- $B_i = -1$ または $B_i = A_i$
  - 協力者がいない（タイプ A）、または票と協力者が同時に得られる（タイプ B）

- このようなケースでは、次の 3 つのことがいえる：
  - タイプ A の中では、簡単な州 [A_i が小さい州] から取った方が良い
  - タイプ B の中では、簡単な州 [A_i が小さい州] から取った方が良い
  - 協力者は早く得た方が良いので、タイプ B の州から演説をすべき

2 小課題 2 の解法

- そこで、タイプ B から取る票の数 $Goal$ を全探索する
  - $Goal$ が決まったら、まずタイプ B の州を A_i の小さい順に $Goal$ 個取る
  - 次に、タイプ A の州を A_i の小さい順に $K - Goal$ 個取る

2 小課題 2 の解法

- そこで、タイプ B から取る票の数 $Goal$ を全探索する
  - $Goal$ が決まったら、まずタイプ B の州を A_i の小さい順に $Goal$ 個取る
  - 次に、タイプ A の州を A_i の小さい順に $K - Goal$ 個取る

	州 1	州 2	州 3	州 4	州 5
票獲得まで A_i	4	5	6	7	8
協力者まで B_i	-	-	-	7	8

左の例で $K = 3$ 票取ることを考える

$Goal = 0$ の場合は…

- 州 $1 \rightarrow 2 \rightarrow 3$ の順に取れば良い
- $4 + 5 + 6 = 15$ 時間かかる

2 小課題 2 の解法

- そこで、タイプ B から取る票の数 $Goal$ を全探索する
  - $Goal$ が決まったら、まずタイプ B の州を A_i の小さい順に $Goal$ 個取る
  - 次に、タイプ A の州を A_i の小さい順に $K - Goal$ 個取る

	州 1	州 2	州 3	州 4	州 5
票獲得まで A_i	4	5	6	7	8
協力者まで B_i	-	-	-	7	8

左の例で $K = 3$ 票取ることを考える

$Goal = 1$ の場合は…

- 州 $4 \rightarrow 1 \rightarrow 2$ の順に取れば良い
- $7 + 4/2 + 5/2 = 11.5$ 時間かかる

2 小課題 2 の解法

- そこで、タイプ B から取る票の数 $Goal$ を全探索する
  - $Goal$ が決まったら、まずタイプ B の州を A_i の小さい順に $Goal$ 個取る
  - 次に、タイプ A の州を A_i の小さい順に $K - Goal$ 個取る

	州 1	州 2	州 3	州 4	州 5
票獲得まで A_i	4	5	6	7	8
協力者まで B_i	-	-	-	7	8

左の例で $K = 3$ 票取ることを考える

$Goal = 2$ の場合は…

- 州 $4 \rightarrow 5 \rightarrow 1$ の順に取れば良い
- $7 + 8/2 + 4/3 = 12.3$ 時間かかる

2 小課題 2 の解法

- そこで、タイプ B から取る票の数 $Goal$ を全探索する
  - $Goal$ が決まったら、まずタイプ B の州を A_i の小さい順に $Goal$ 個取る
  - 次に、タイプ A の州を A_i の小さい順に $K - Goal$ 個取る

$Goal = 0, 1, 2$ の中で最も時間が少ない

11.5 時間が答え

	州 1	州 2	州 3	州 4	州 5
票獲得まで A_i	4	5	6	7	8
協力者まで B_i	-	-	-	7	8

の 5 州で $K = 3$ 票取ることを考える

$Goal = 2$ の場合は…

- 州 $4 \to 5 \to 1$ の順に取れば良い
- $7 + 8/2 + 4/3 = 12.3$ 時間かかる

2 小課題 2 の実装


```
#include <bits/stdc++.h>
using namespace std;
int N, K, A[509], B[509];
vector<int> vec1, vec2;

int main() {
    // 入力
    cin >> N >> K;
    for (int i = 0; i < N; i++) {
        cin >> A[i] >> B[i];
        if (B[i] == -1) vec1.push_back(A[i]);
        if (B[i] == A[i]) vec2.push_back(A[i]);
    }

    // ソート
    sort(vec1.begin(), vec1.end());
    sort(vec2.begin(), vec2.end());

    // 答えを求める
    double Answer = 1e9;
    for (int Goal = 0; Goal <= K; Goal++) {
        if (K - Goal > vec1.size() || Goal > vec2.size()) continue;
        double sum = 0;
        for (int i = 0; i < K - Goal; i++) sum += 1.0 * vec1[i] / (Goal+1);
        for (int i = 0; i < Goal; i++) sum += 1.0 * vec2[i] / (i+1);
        Answer = min(Answer, sum);
    }
    printf("%.12lf¥n", Answer);
    return 0;
}
```

- したがって、左のような実装をすれば良い
- $Goal$ の値を $O(N)$ 通り全探索しているので、計算量は $O(N^2)$
- $N \leq 500$ なので間に合う

10点

小課題 3

$N \leq 7$

3 小課題 3 の解法

- 各州について、最適解は明らかに以下の 3 つのいずれか：
  - パターン A：0 時間演説する
  - パターン B：A_i 時間演説する（票を得る）
  - パターン C：B_i 時間演説する（協力者を得る）

- 各州の演説時間が決まれば、最適なムーブはすぐに分かる
  - 小課題 2 と同様、協力者を得てから票を得るのが最適

3 小課題 3 の解法

- そこで、各州の演説時間を全探索することを考える
  - 全部で $3 \times 3 \times \cdots \times 3 = 3^N$ 通りある
  - $N \leq 7$ なので、小課題 3 が通る

- 制約が小さいため、「協力者を得てから票を得る」という考察をせずに演説の順番を全探索しても通る

21点

小課題 4

$N \leq 20$

4 小課題 4 の解法

- 「どの州で協力者を得るか？」を全探索することを考える
  - 全部で 2^N 通りを調べる必要がある

- 協力者を得る州が決まったら…
  - あとは明らかに A_i の小さい順に票を得れば良い
  - したがって、「0 時間か A_i 時間か」を全探索せずとも演説をする州が決まる

4 小課題 4 の解法

たとえば以下の例を考えよう（$K = 4$ 票得れば良い）

	州 1	州 2	州 3	州 4	州 5	州 6
票獲得まで A_i	3時間	5時間	6時間	4時間	7時間	2時間
協力者まで B_i	3時間	6時間	6時間	8時間	9時間	12時間

4 小課題 4 の解法

たとえば以下の例を考えよう（$K = 4$ 票得れば良い）

	州 1	州 2	州 3	州 4	州 5	州 6
票獲得まで A_i	3時間	5時間	6時間	4時間	7時間	2時間
協力者まで B_i	3時間	6時間	6時間	8時間	9時間	12時間

州 1 と州 3 で協力者を得たとする

4 小課題 4 の解法

たとえば以下の例を考えよう（$K = 4$ 票得れば良い）

	州 1	州 2	州 3	州 4	州 5	州 6
票獲得まで A_i	3時間	5時間	6時間	4時間	7時間	2時間
協力者まで B_i	3時間	6時間	6時間	8時間	9時間	12時間

州 1 と州 3 で協力者を得たとする

- 残り 2 票は、A_i の値が小さい州 4 と州 6 で演説をすれば良い
- 合計時間は $3 + 6/2 + 4/3 + 2/3 = 8$ 時間

4 小課題 4 の解法

たとえば以下の例を考えよう（$K = 4$ 票得れば良い）

	州 1	州 2	州 3	州 4	州 5	州 6
票獲得まで A_i	3時間	5時間	6時間	4時間	7時間	2時間
協力者まで B_i	3時間	6時間	4時間	8時間	9時間	12時間

ビット全探索などを用いて「協力者を得る州」を全て調べれば良い！

計算量は $O(2^N \times N \log N)$

州 1 と州 3 で協力者を得ることを考える

- 残り 2 票は、A_i の値が小さい州 4 と州 6 で演説をすれば良い
- 合計時間は $3 + 6/2 + 4/3 + 2/3 = 8$ 時間

※ここまで 33 点

小課題 5

$N \leq 100$

5 小課題 5 の解法

小課題 2 の解法を

思い出してみましょう

5 小課題 5 の解法

- 小課題 2 では、最終的な協力者の人数 $Goal$ を全探索した
  - 整数 $Goal$ が決まった場合のメリットは…？
  - 「票しか得ない州」に必要な演説時間も定まる

- A_i, B_i が一般のケースでも、$Goal$ の値が決まると DP とかに落とせたりしないのか？

5 小課題 5 の解法

以下のアルゴリズムにより、計算量 $O(N^4)$ が実現できる

手順 1　協力者を得るまでにかかる時間 B_i の昇順にソートする

手順 2　協力者の人数 $Goal$ を決めた上で、次のような DP をする

- dp[何番目の州まで決めたか][得た票の数][協力者の人数] = 時間の最小値
- 遷移は、州の演説時間が $\{0, A_i, B_i\}$ のどれであるかを一つずつ決めていく感じ

5 小課題 5 の解法

以下のアルゴリズムにより、計算量 $O(N^4)$ が実現できる

手順 1　協力者を得るまでにかかる時間 B_i の昇順にソートする

手順 2　協力者の人数 $Goal$ を決めた上で、次のような DP をする

- dp[何番目の州まで決めたか][得た票の数][協力者の人数] = 時間の最小値
- 遷移は、州の演説時間が $\{0, A_i, B_i\}$ のどれであるかを一つずつ決めていく感じ

5 小課題 5 の解法

以下のアルゴリズムにより、計算量 $O(N^4)$ が実現できる

手順 1　協力者を得るまでにかかる時間 B_i の昇順にソートする

手順 2　協力者の人数 $Goal$ を決めた上で、次のような DP をする

- dp[何番目の州まで決めたか][得た票の数][協力者の人数] = 時間の最小値
- 遷移は、州の演説時間が $\{0, A_i, B_i\}$ のどれであるかを一つずつ決めていく感じ

5 小課題 5 の解法

以下のアルゴリズムにより、計算量 $O(N^4)$ が実現できる

手順 1　協力者を得るまでにかかる時間 B_i の昇順にソートする

手順 2　協力者の人数 $Goal$ を決めた上で、次のような DP をする

- dp[何番目の州まで決めたか][得た票の数][協力者の人数] = 時間の最小値
- 遷移は、州の演説時間が $\{0, A_i, B_i\}$ のどれであるかを一つずつ決めていく感じ

演説をしない場合	$\mathrm{dp}[i+1][j][k] \coloneqq \mathrm{dp}[i][j][k]$	
票だけ取る場合	$\mathrm{dp}[i+1][j+1][k] \coloneqq \mathrm{dp}[i][j][k] + A_i/(Goal+1)$	← 票だけ取る州は最後に取る
協力者まで取る場合	$\mathrm{dp}[i+1][j+1][k+1] \coloneqq \mathrm{dp}[i][j][k] + B_i/(k+1)$	← 協力者は B_i の小さい順に取る

5 小課題 5 の解法

- 計算量を考えてみよう
  - 協力者の人数 $Goal$ は $O(N)$ 通り全探索する
  - DP は三次元配列を使うので、状態数が $O(N^3)$ 通り

- 全体の計算量は $O(N) \times O(N^3) = O(N^4) \rightarrow N \leq 100$ なら間に合う
  - 過半数獲得おめでとう！

小課題 6

$N = K$

6 小課題 6 の解法

- $N = K$ なので、「0 時間演説する」という選択肢はない
  - そのため、暫定獲得票数は DP 配列にメモしなくて良い

- 協力者の人数 x を全探索したうえで、次のような DP をすれば良い
  - dp[何番目の州まで決めたか][協力者の人数] = 時間の最小値
  - ただし、小課題 5 と同様 B_i の昇順にソートする必要がある

6 小課題 6 の解法

- 実装は小課題 5 を少し変更するだけで良い
  - DP の「暫定獲得票数」の次元を削り、票を得ない選択肢を消すだけ

- 計算量は $O(N^3)$ なので、$N \leq 500$ でも間に合う
  - 3 分の 2 獲得おめでとう！

小課題 7

追加の制約はない

7 小課題 7 の解法

- 簡単のため、$B_1 \leq B_2 \leq \cdots \leq B_N$ を仮定する
- そのとき、次のような組 $x < y$ があるのは絶対に損
  - 州 x で演説をせず、州 y で B_y 時間演説して協力者を得る

7 小課題 7 の解法

- 簡単のため、$B_1 \leq B_2 \leq \cdots \leq B_N$ を仮定する
- そのとき、次のような組 $x < y$ があるのは絶対に損
  - 州 x で演説をせず、州 y で B_y 時間演説して協力者を得る

なぜか？ → 州 y の代わりに州 x で協力者を得た方が時間が短い！

	州1	州2	州3	州4	州5
票獲得まで A_i	1	6 ←	4	4	8
協力者まで B_i	3	6 ←	7	9	11

→

	州1	州2	州3	州4	州5
票獲得まで A_i	1	6	4	4	8
協力者まで B_i	3	6	7	9	11

※青は票獲得だけ、赤は協力者まで

7 小課題 7 の解法

- 簡単のため、$B_1 \leq B_2 \leq \cdots \leq B_N$ を仮定する
- そのとき、次のような組 $x \leq y$ があるのは絶対に損
  - 州 x で演説をせず、州 y で B_y 時間演説して協力者を得る

州 1 → 2 → 3 → … → N の順に決めていったとき

協力者を目標の x 人集めるまでは

なぜか？ → 州 y の代わりに州 x で協力者を得た方が時間が短い！

演説しないという選択肢はない！

	州1	州2	州3	州4	州5
票獲得まで A_i	1	6	4	4	8
協力者まで B_i	3	6	7	9	11

➡

	州1	州2	州3	州4	州5
票獲得まで A_i	1	6	4	4	8
協力者まで B_i	3	6	7	9	11

※青は票獲得だけ、赤は協力者まで

7 小課題 7 の解法

そこで小課題 5 の解法を振り返ってみよう

手順 1　協力者を得るまでにかかる時間 B_i の昇順にソートする

手順 2　協力者の人数 $Goal$ を決めた上で、次のような DP をする

- dp[何番目の州まで決めたか i][得た票の数 j][協力者の人数 k] = 時間の最小値
- 遷移は、州の演説時間が $\{0, A_i, B_i\}$ のどれであるかを一つずつ決めていく感じ

7 小課題 7 の解法

そこで小課題 5 の解法を振り返ってみよう

手順 1　協力者を得るまでにかかる時間 B_i の昇順にソートする

手順 2　協力者の人数 $Goal$ を決めた上で、次のような DP をする

- dp[何番目の州まで決めたか i][得た票の数 j][協力者の人数 k] ＝ 時間の最小値
- 遷移は、州の演説時間が $\{0, A_i, B_i\}$ のどれであるかを一つずつ決めていく感じ

➡ 協力者の人数が $Goal - 1$ 以下の場合は、$i = j$ が成り立つ

※ $i \neq j$ の場合、協力者が全員集まる前に「演説しないこと」を選んでしまう！

7 小課題 7 の解法

- 考えるべき $dp[i][j][k]$ の状態数はいくつあるのか？
  - i: 何番目の州まで決めたか（$0 \leq i \leq N$）
  - j: 得た票の数（$0 \leq j \leq K$）
  - k: 得た協力者の人数（$0 \leq k \leq Goal$、$Goal$ は最終的な協力者数）

- 実は全部で $O(N^2)$ 通りしか考える必要がない
  - $k = Goal$ のとき：全体で高々 $(N+1)(K+1)$ 通り
  - $k < Goal$ のとき：$i = j$ なので各 k $(0 \leq k < Goal)$ につき高々 N 通り

7 小課題 7 の実装

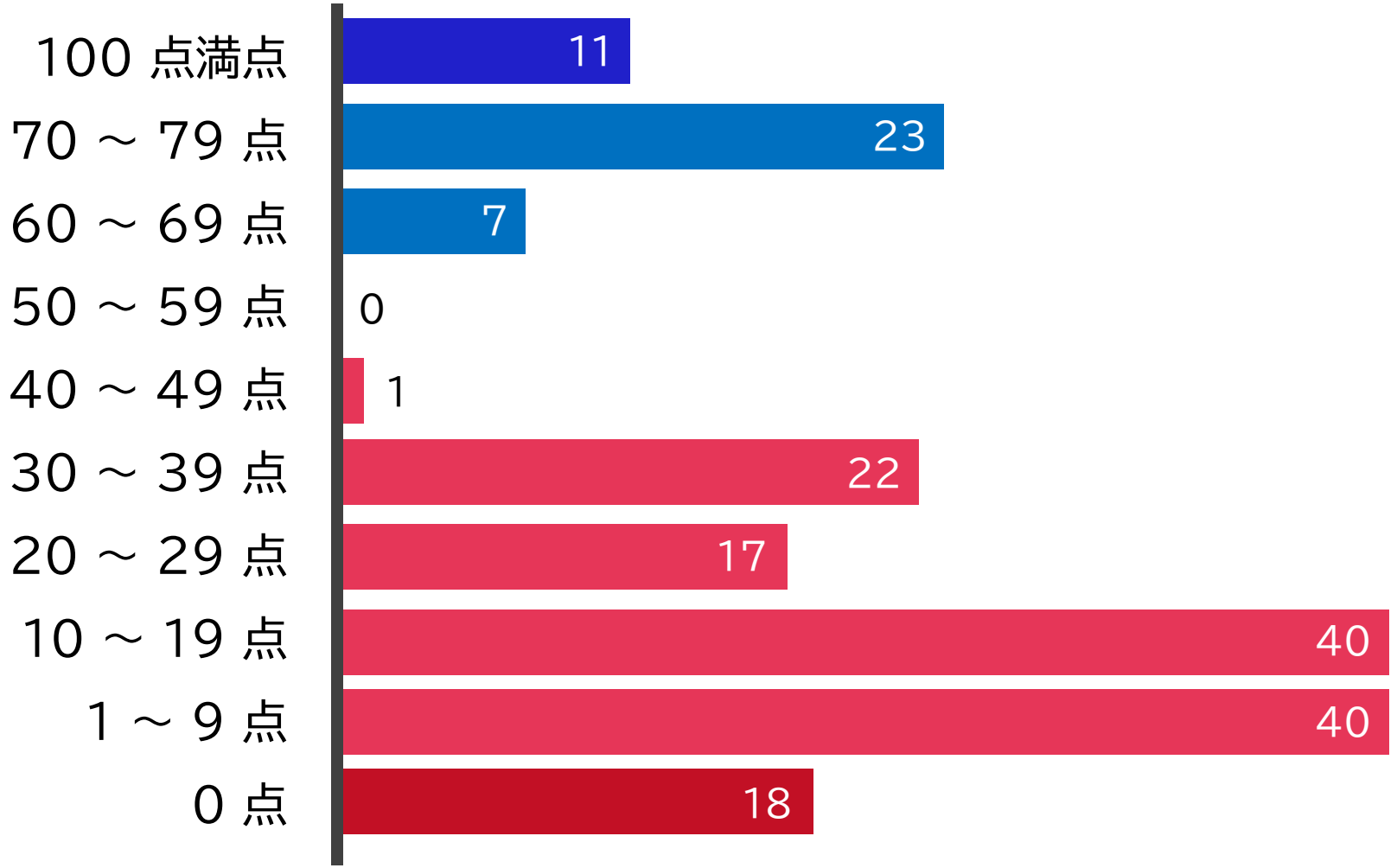
- $Goal$ を $0 \leq Goal \leq N$ の範囲で全探索するので、計算量は $O(N^3)$
  - 上手に実装すれば $N \leq 500$ でも通る
  - 全議席獲得おめでとう！

- サンプルコードはメモリ使用量 $O(N^3)$ の実装となっていますが、DP 配列を 2 つ用意するなどすれば、$O(N^2)$ に減らすことも可能です

100点

得点分布

8 得点分布



50点以上 41

50点未満 138

平均点 28.5