

ABC 384 C - Perfect Standings

hiragn

2024 年 12 月 16 日

1. 問題の概要

プログラミングコンテストは A 問題～E 問題の 5 問からなり、配点は $a \sim e$ 点である。コンテストには 31 人が参加し、全員が 1 問以上解いた。

たとえば A さんは A 問題のみを解き、BCE さんは B, C, E の 3 問を解いた。このように名前と解いた問題の組は一致している。

参加者の名前を点数が大きい方から順に出力せよ。同点の場合は名前が辞書順で小さい方を先に出力せよ。

https://atcoder.jp/contests/abc384/tasks/abc384_c

2. 解法

2.1 得点

1～31 を 5 桁の 2 進数に変換して、その桁数字のベクトル \vec{v} を考える。得点はこれと配点ベクトル $\vec{p} = (a, b, \dots, e)$ の内積で求められる。

たとえば入力例 1 の場合 $\vec{p} = (400, 500, 600, 700, 800)$ であり、11 ($= 0 + 8 + 0 + 2 + 1$) の人は B, D, E を正解している。その得点は 2000 点である。

$$(0, 1, 0, 1, 1) \cdot \vec{p} = 500 + 700 + 800 = 2000 \text{ 点}$$

```
1 In[] := n = 11;
2 p = {400, 500, 600, 700, 800};
3 v = IntegerDigits[n, 2, 5];
4 score = v . p
5
6 Out[] = 2000
```

2.2 名前

名前を作るには Position で \vec{n} の 1 のインデックスを拾って、FromLetterNumber に渡せばよい。StringJoin で連結して、ToUpperCase で大文字に直す。

```

1 In[] := v = IntegerDigits[n, 2, 5];
2 ToUpperCase@StringJoin[FromLetterNumber @@@ Position[v, 1]]
3
4 Out[] = "BDE"

```

2.3 ソート

得点については降順、名前については昇順にソートする。ソートの条件が 2 つあるので SortBy を使う。ソート順はデフォルトで昇順なので得点を -1 倍して使う。これで完成。

```

1 In[] := Clear["Global`*"];
2 solve[points_] := Module[{score, name},
3   score[dgts_] := -points . dgts;
4   name[dgts_] := name[dgts] =
5     ToUpperCase@StringJoin[FromLetterNumber @@@ Position[dgts, 1]];
6   ans = name /@ SortBy[
7     IntegerDigits[#, 2, 5] & /@ Range@31, {score, name}]];
8
9 case1 = {400, 500, 600, 700, 800};
10 case2 = {800, 800, 900, 900, 1000};
11 case3 = {128, 256, 512, 1024, 2048};
12 res1 = {"ABCDE", "BCDE", "ACDE", "ABDE", "ABCE", "ABCD", "CDE", "BDE",
13   "ADE", "BCE", "ACE", "BCD", "ABE", "ACD", "ABD", "ABC", "DE",
14   "CE", "BE", "CD", "AE", "BD", "AD", "BC", "AC", "AB", "E", "D",
15   "C", "B", "A"};
16 res2 = {"ABCDE", "ACDE", "BCDE", "ABCE", "ABDE", "ABCD", "CDE", "ACE",
17   "ADE", "BCE", "BDE", "ABE", "ACD", "BCD", "ABC", "ABD", "CE",
18   "DE", "AE", "BE", "CD", "AC", "AD", "BC", "BD", "AB", "E", "C",
19   "D", "A", "B"};
20 res3 = {"ABCDE", "BCDE", "ACDE", "CDE", "ABDE", "BDE", "ADE", "DE",
21   "ABCE", "BCE", "ACE", "CE", "ABE", "BE", "AE", "E", "ABCD", "BCD",
22   "ACD", "CD", "ABD", "BD", "AD", "D", "ABC", "BC", "AC", "C", "AB",
23   "B", "A"};
24 solve /@ {case1, case2, case3} == {res1, res2, res3}
25
26 Out[] = True

```
