

ICPC 模擬国内予選 2017

A: JAG 模擬予選練習会

原案: 井上 (darsein)

問題文: 山崎 (torus711)

データセット: 杉田 (kurome)

解答: 飯塚 (shora_kujira16)

杉田

水野 (not_522)

解説: 杉田

問題概要

- プログラミングコンテストの問題が N 問用意されている
- i 番目の問題には, その問題の「難易度」 $d[i]$ と「推薦度」 $v[i]$ が設定されている
- 全 N 問から, 難易度が $1, 2, \dots, M$ の問題を1問ずつ適切に選んで, 選んだ M 問の推薦度の合計が取りうる最大値を求めて欲しい
- 制約:
 - $1 \leq N \leq 100$
 - $1 \leq M \leq 100$
 - $1 \leq d[i] \leq M$ ($1 \leq i \leq N$)
 - $0 \leq v[i] \leq 100$ ($1 \leq i \leq N$)
 - 用意された N 問には, 1 から M までの難易度の問題がそれぞれ少なくとも1問は存在することが保証されている

想定解法1

- 各難易度 i ($1 \leq i \leq M$)に対して、以下の処理を繰り返す

難易度が i である問題を用意された N 問から線形探索し、該当する問題の推薦度の最大値を求める

- 難易度毎に得られた推薦度の最大値の合計が答え

上記の実装コード例 (C++) :

```
int total = 0;
for (int i=1; i<=M; i++) {
    int max_v = 0;
    for (int j=1; j<=N; j++) {
        if (d[j] == i) max_v = max(max_v, d[j]);
    }
    total += max_v;
}
```

```
cout << total << endl;
```

- 計算量 $O(MN)$ (*別途入力データの取得に $O(N)$ 必要)

想定解法2

- 難易度毎に、その難易度を持つ問題の中で最大の推薦度を持たせる配列を用意
例えば、最終的に $\text{max_v}[i] = \max\{\text{難易度が } i \text{ の問題の推薦度}\}$ となるように $\text{max_v}[i]$ ($1 \leq i \leq M$) を用意
- 用意された各問題 i ($1 \leq i \leq N$) に対して、 $\text{max_v}[d[i]]$ を、 $v[i]$ との最大値をとって更新する処理を繰り返す
- $\text{max_v}[1]$ から $\text{max_v}[M]$ までの和が答え

上記の実装コード例 (C++) :

```
for (int i=1; i<=N; i++) {  
    max_v[d[i]] = max(max_v[d[i]], v[i]);  
}
```

```
int total = 0;  
for (j=1; j<=M; j++) {  
    total += max_v[j];  
}
```

```
cout << total << endl;
```

- 計算量: $O(N)$ (※別途入力データの取得にも $O(N)$ 必要)

ジャッジ解

- 解法1

水野(C++): 27行, 539バイト

- 解法2

飯塚 (python): 16行, 294バイト

杉田 (C++): 33行, 596バイト

水野(C++): 25行, 474バイト

※コメント等を含む

統計情報

- Accepted

128 チーム

- First acceptance

- 全チーム内

RabbitHose (00:05:45)

- 現役チーム内

tabanaka (00:06:52)