



Problem B:

Moonlight Farm

Slides by Yusuke Izumi

問題概要

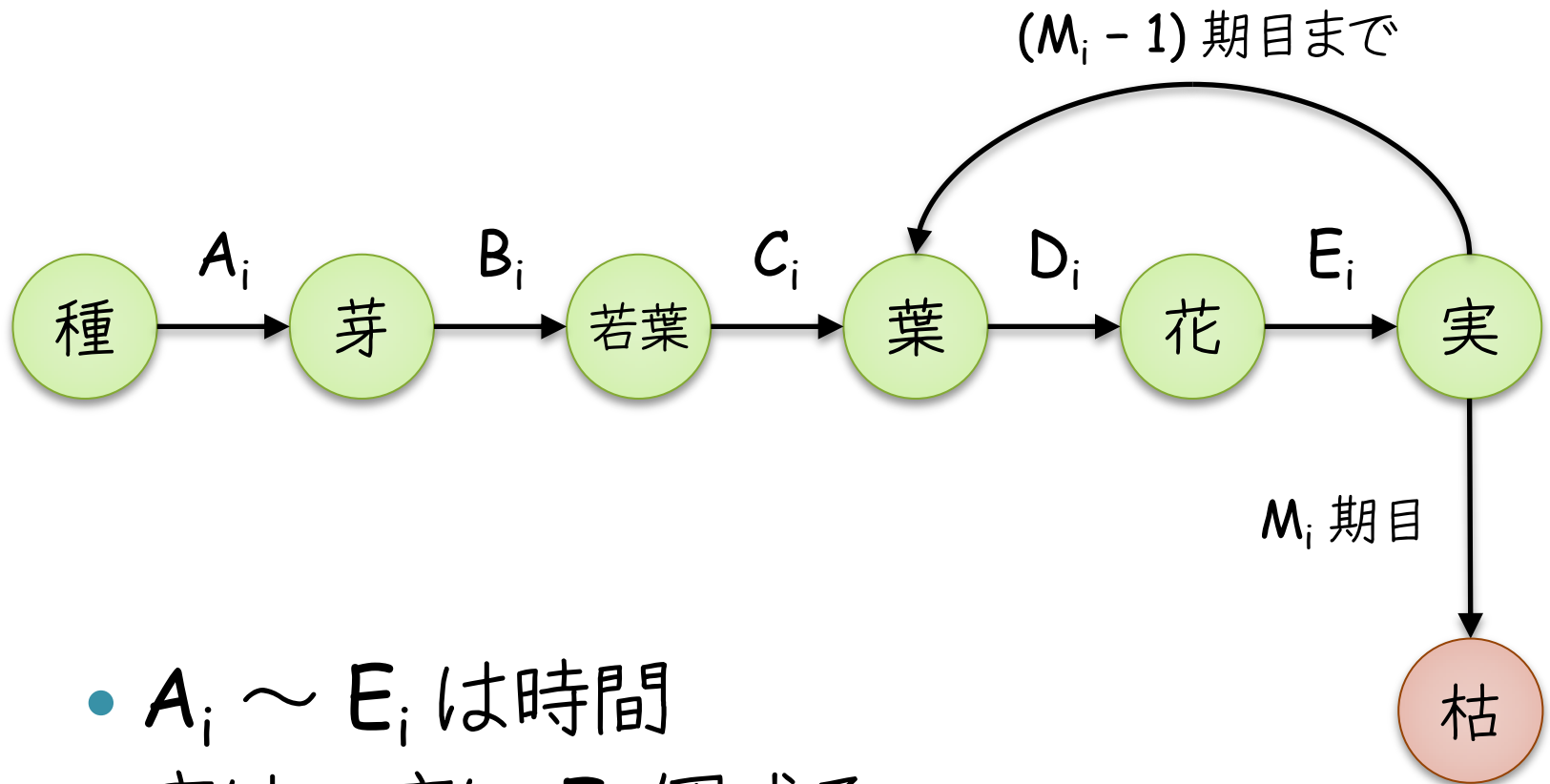
- 作物の一覧が与えられる.
- 作物を「収入効率」の逆順に並べる.

$$\text{収入効率} = \frac{\text{ひとつの種から得られる全収入}}{\text{最後の実が成るまでにかかる時間}}$$

解法

- 問題文をよく読んで整理する.
- 無駄な情報に振り回されない.

作物の周期



- $A_i \sim E_i$ は時間
- 実は一度に F_i 個成る
- 実は 1 個あたり S_i で売れる

作物の収入効率

- 収入
 - 一周期ごとの売値は $S_i F_i$.
 - これが M_i 周期あって、種の価格が P_i なので、収入は $S_i F_i M_i - P_i$.
- 時間
 - 種から葉になるまで $A_i + B_i + C_i$.
 - 葉から実が成るまで $D_i + E_i$.
 - 種から葉までの過程は 1 回だけで、葉から実までの過程は M_i 回ある.
 - 全体では $A_i + B_i + C_i + (D_i + E_i) \times M_i$.

まとめると

- 作物を以下の基準に沿って並べ替える。
 - 収入効率の降順

$$\frac{S_i F_i M_i - P_i}{(A_i + B_i + C_i) + (D_i + E_i) \times M_i}$$

- 名前の昇順
- 構造体のソートが必要になる。

構造体のソート

- C++

- `operator <` を再定義する.
- 比較関数を定義して `sort()` 関数に渡す.

- Java

- 元のクラスに `Comparable` インターフェイスを実装する.
- `Comparator` インターフェイスを実装したクラスを用意して `sort()` メソッドに渡す.

収入効率の比較

- 分数として比較する.
 - $(a / b) < (c / d) \Leftrightarrow (a * d) < (b * c)$
- 実数として比較する.
 - 単純に $a < b$ としないこと.
 - たとえば $(a - b) < 0.000000001$ とする.

実装例(C++)

```
struct Crop {
    string name;
    int profit; // 合計収入
    int period; // 合計時間

    bool operator <(const Crop& o) const {
        if (profit * o.period < period * o.profit)
            return true;
        if (profit * o.period > period * o.profit)
            return false;
        return name < o.name;
    }
};

vector<Crop> crops;
sort(crops.begin(), crops.end());
```

実装例 (Java)

```
class Crop implements Comparable<Crop> {
    string name;
    int profit;    // 合計収入
    int period;   // 合計時間

    public int compareTo(Crop o) {
        if (profit * o.period < period * o.profit)
            return -1;
        if (profit * o.period > period * o.profit)
            return +1;
        return name.compareTo(o.name);
    }
}

Crop[] crops;
Arrays.sort(crops);
```

統計情報

- 提出数 : 412 (?? teams)
- 正答数 : 96
- 最初の提出 : 9.5 mins
- 最初の正答 : 17 mins (Imo)

クレジット

- 原案
 - 野田
- 文章
 - 野田
- 解答
 - 北村, 泉
- 入力
 - 北村