

AtCoder Beginner Contest 019

解説



AtCoder株式会社 代表取締役
高橋 直大

-
- 競技プログラミングをやったことがない人へ
 - まずはこっちのスライドを見よう！
 - <http://www.slideshare.net/chokudai/abc004>

A問題 高橋くんと年齢

1. 問題概要
2. アルゴリズム

- 3つの整数が与えられる.
- それらの中央値を求めよ.
- 制約
 - $1 \leq (\text{与えられるそれぞれの整数}) \leq 1000$

- 解法1

- 3つの整数を配列にしてソートします.

- 2番目の値を答えます.

- C++ なら, `int ar[3]={a,b,c}; sort(ar,ar+3);` とし, `ar[1]` が答え

- また, 以下のように $a \leq b \leq c$ と並べて, `b` を答えにすることもできます

- ```
if(a>b) swap(a,b);
```

- ```
if(b>c) swap(b,c);
```

- ```
if(a>b) swap(a,b);
```

- 解法2

- 3つの整数のうち, 最大値でも最小値でもないものを探してその値を答えます.

## B問題 高橋くんと文字列圧縮

---

1. 問題概要
2. アルゴリズム

- 英小文字からなる文字列  $s$  が与えられる.
- $s$  を連続する文字とその長さをつなげた文字列に変換せよ.
- 制約
  - $0 \leq |s| \leq 100000$

- 
- 問題文通りに文字列を分解, 文字と長さへ変換してつなげて大丈夫です.
  - 数字を文字列に変換するには, C++なら `stringstream`, Java なら `String.valueOf` が便利です.



- その他に、最初の文字から続く部分を文字と長さへ変換し、答えの文字列にくっつけて、 $s$ の次の文字へ進む、という処理を繰り返す方法もあります。

– 分解とかをしないので楽かも。答えの文字列を  $t$  とおくと、

```
int seek=0;
while(seek<|s|){
 int startPos=seek;//開始位置
 while(seek<n && s[seek]==s[startPos]) ++seek;
 //同じ文字がどこまで続くか探す
 t+=s[startPos];
 t+=itos(seek-startPos);
}
```

(itosは整数から文字列への変換関数とします)

## C問題 高橋くんと魔法の箱

---

1. 問題概要
2. アルゴリズム

- 魔法の箱に整数を入れると、それに対応した整数が出てくる。
- 任意の整数  $x$  について、 $x$  を入れた時に出てくる整数は、 $2x$  の時と等しい
- $N$  個の整数  $a_i$  を魔法の箱に入れた時、最大何種類の整数が得られるか。
- 制約
  - $1 \leq N \leq 10^5, 1 \leq a_i \leq 10^9$
  - 20点:  $1 \leq N \leq 3000$
  - 50点:  $1 \leq a_i \leq 500000$

- ある数  $x$  に対する結果と  $2x$  に対する結果が同じなので,  $a_i$  が2で割り切れなくなるまで2で割り続けてもよい.
- この処理で,  $a_i$  がすべて奇数になった問題にすることができる. 逆に, 奇数で異なる数に対する答えをまとめて求めることは不可能.
- よって, あとは異なる  $a_i$  の数を求めればよい.

- 20点解法

- $n \leq 3000$  なので,  $a_i$  が  $a_j$  ( $1 \leq j < i$ ) のいずれかに等しいかをすべてチェックしてもOK.
- いずれにも等しくないとき,  $a_i$  の種類を1増やす.

- 30点解法

- $a_i \leq 300000$  なので, ある数が現れたかどうか, という長さ300000の配列を取り, それぞれの  $a_i$  について, 現れたことをメモする.
- 最後に現れたことのある数を数える.

- 合わせて50点

- `if(N<=3000) solve20();`  
`else solve30();` のようにして, 1つの提出の中に組み込む.

- 満点解法では,  $N$ 個の数の中から異なる数を効率的に数えなければならない.
- `std::set(C++)` や `HashSet(Java)` を使う解法
  - これらは要素の重複を許可しないので, 同じ値が来ると自動で捨ててくれる
  - $a_i$  をこれらのデータ構造に詰め込んで, 最終的な要素の数が答え.
- ソートを使う方法
  - $a_i$  をソートし, 初めから見っていく. ある値を見つけたら, 答えを1増やし, そこから連続した同じ値の場所をスキップして, 次の値の場所へ移動, を繰り返す(問題Bと似てる)

## D問題 高橋くんと木の直径

---

1. 問題概要
2. リアクティブについて
3. アルゴリズム

- $N$ 頂点からなる重み付き根なし無向木がある.
- $N$ が与えられ, 以下の質問を何回かできる
  - 頂点 $a$ と頂点 $b$ の間の距離を尋ねる
- 木の直径を答えよ.
- 制約
  - $2 \leq N \leq 50$
  - 20点: 質問回数  $\leq 1300$
  - 80点: 質問回数  $\leq 100$



- この問題はリアクティブ形式の問題です.
  - 滅多にみないかもしれませんが KUPC・IOI などではよく出ます
  - デバッグが少し大変になりがちです(基本的に手元では返答を手で行う必要があるので)
    - たくさん問題があるコンテストで実装の重いリアクティブは後回しにするのも手

- 
- 20点解法では, 単純にすべての2頂点のペアについて, その間の距離を尋ね, 最大値が直径となる.
  - 頂点aと頂点bの間の距離と頂点bと頂点aの間の距離が同じなので,  $a < b$  なもののみ聞けばよい.
  - 質問回数は  $N * (N - 1) / 2 \leq 1300$

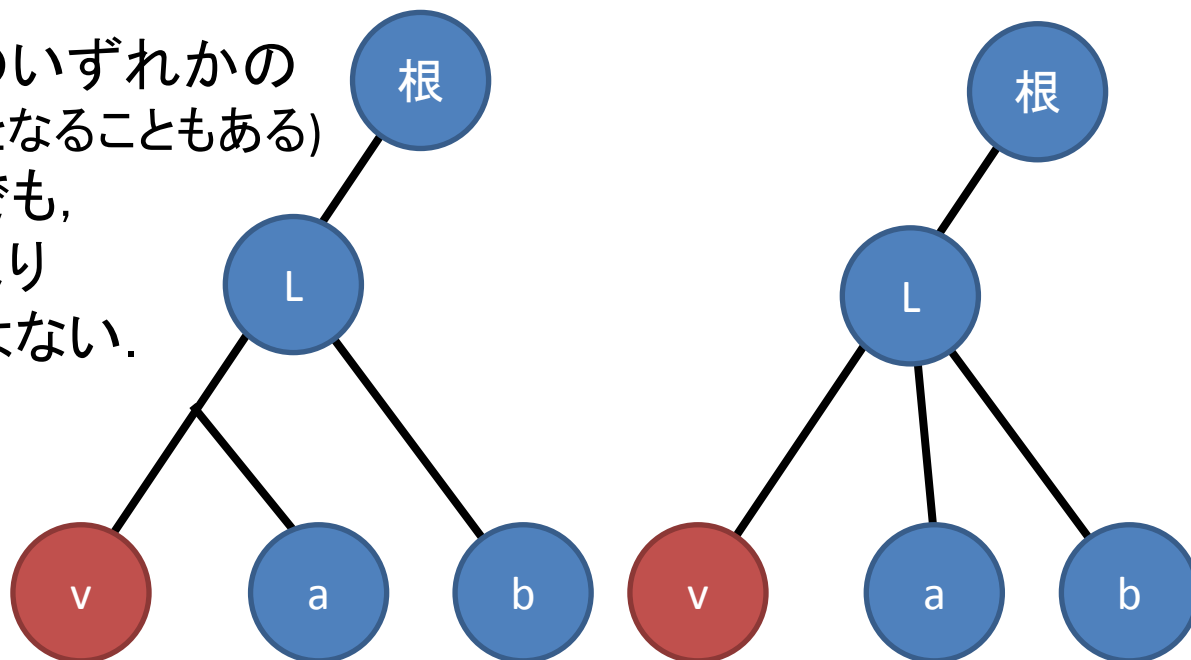
- 最大ケースだと、満点解法では $2N$ 回以下の質問回数で直径を求めなければならない。
- 辺を特定するのは現実的ではなさそう
  - ある頂点から一番近い頂点へは直接辺でつながっていると分かるが、 $N-1$ 回の質問で最悪1つの辺しか見つけられない
- 辺を特定することなく直径だけ求める方法をうまいこと考えたい/探したい。

- 木の直径を求める方法の1つに double-sweep という方法があります.
- 木の頂点から任意の1つの頂点から一番離れた頂点を  $v$  とすると,  $v$  から最も離れた点までの距離が直径になる, というもの.
- 何故か?
- (以後2つの頂点  $a, b$  の間の距離を  $a--b$  と書きます)

- まず, 任意の1つの頂点を根と決める. そこから最も離れている頂点のうち1つを  $v$  とおく.  $v$  を片方の頂点として持つ, 距離が直径のペアがあることを示したい.
- 背理法を使う.  $v$  と  $v$  から最も離れた点への距離より長い頂点の組を  $a, b$  とおく.
- $a$  と  $b$  のLCA (最近共通祖先,  $L$  とおく) が根と  $v$  の間にあるとき

- 右の2つの図のいずれかの形を取る( $L$ =根となることもある)
- どちらの場合でも,  $v$ -- $b$  は  $a$ -- $b$  より短くなることはない.

$\because v$  は  $L$  から最も離れた頂点のうちの一つであることに注目



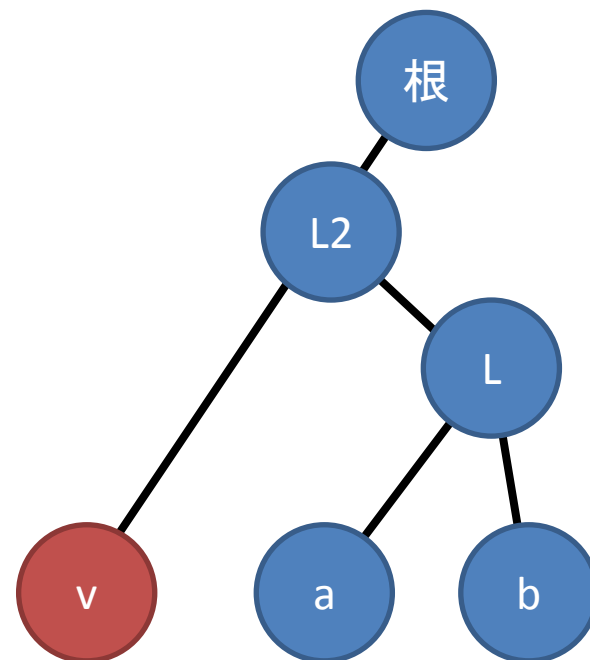
- aとbのLCA(最近共通祖先, Lとおく)が根とvの間でないとき

- LとvのLCAをL2とおくと, 右下のような図になる.
- この場合では, v--bの間の距離はa--bの距離より長い
  - vはLから最も離れた頂点のうちの1つなので,

$$L2-v \geq L2-L + L-a$$

- よって,

$$\begin{aligned} v-b &= L2-v + L2-L + L-b \geq 2 * L2-L + L-a + L-b \\ &> L-a + L-b \geq a-b \end{aligned}$$



- よって、任意に1つの頂点を取った時に、そこから最も離れた頂点のうち1つを使って直径が取れることが分かった.
- この問題なら、頂点1から他のすべての頂点への距離を尋ねて最も遠い頂点を求め、そこからさらに他のすべての頂点への距離を尋ね、その最大値が答え. 質問回数は  $2*N-2$
- 木の直径を求める時はこの知識を知らなくても大丈夫だが、知っていると役に立つときは他にもあつたりします.