ACPC2020 Day3 D
解説

原案 : Reider
問題文 : Reider
解説 : pitsu

問題概要

初め、整数1を持っている。

N個の整数を用いて持っている整数xを(x × A_i)modMに変換する。

0以上、M-1以下の全ての整数を作ることが出来るか判定せよ。

解法

この問題の考察stepは大きく分けて以下の3つのパートがあります。

・0以上、M-1以下の整数を頂点とみなし、グラフ問題にする

・強連結成分分解をする
※強連結成分分解についての説明は省きます

・強連結成分分解後の頂点の集合を1つの頂点として全ての頂点を
たどる方法があるかDPをする（最長経路問題）

解法（グラフ問題にする）

0以上M-1以下の整数を頂点とみなしてグラフ問題にします。

例として

3 8

2 3 5

が入力で与えられたものを考えてみましょう。

解法（グラフ問題にする）

M ＝ 8なので頂点は0以上7以下の8個です。

整数xを用いたとき、

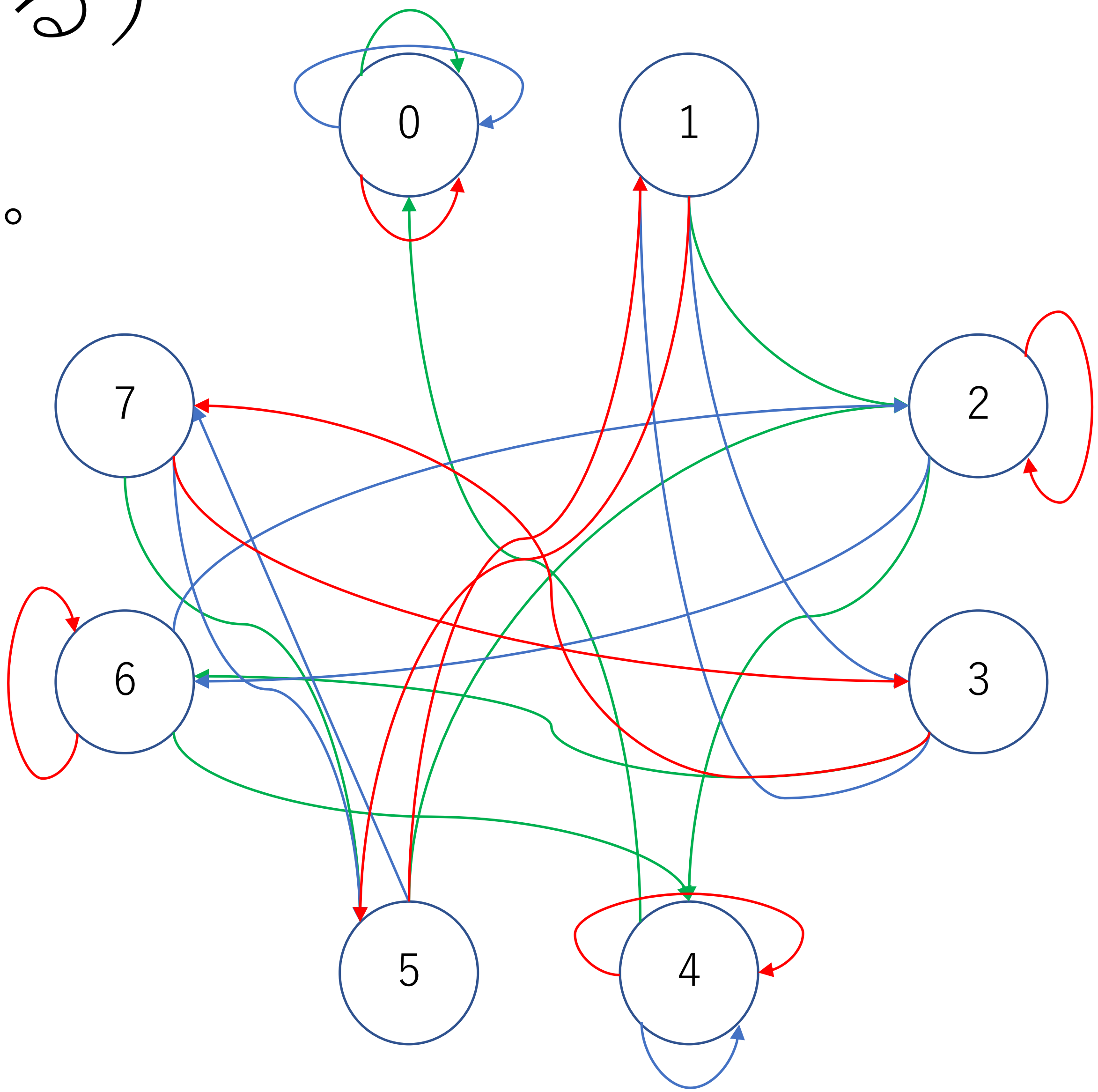
頂点vから頂点(v × x)modMへの
有向辺が張られます。

2によって張られた辺を緑色

3によって張られた辺を青色

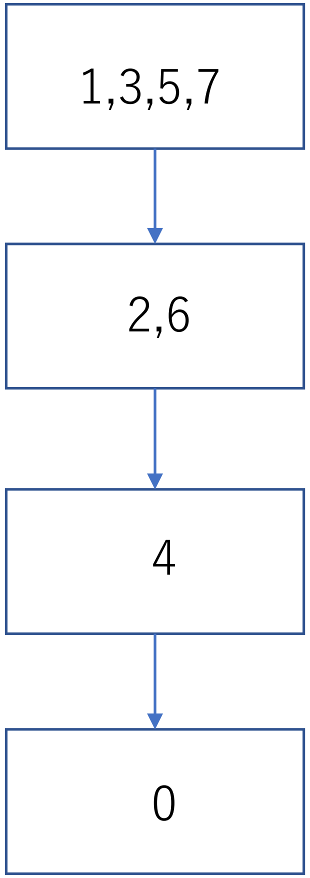
5によって張られた辺を赤色

で表しています。



解法（強連結成分分解）

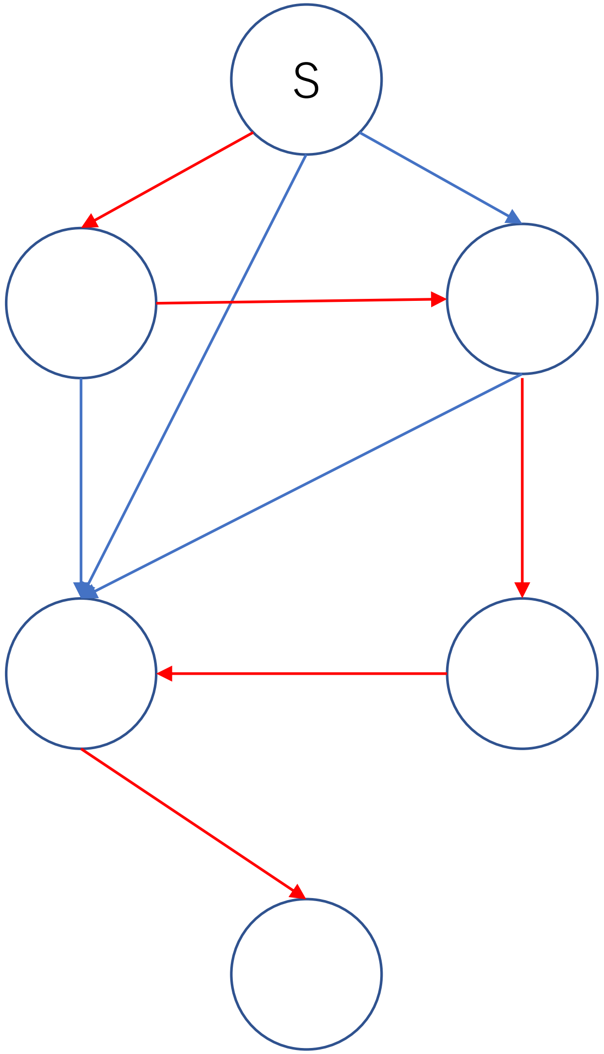
先ほどのグラフを強連結成分分解すると
右のようなグラフになります。
強連結成分分解後の頂点の集合内では
互いに行き来することができます。
そのため、頂点1を含む集合からスタートして
全集合をたどることはできるか？という問題に
言い換えられます。



解法（最長経路問題）

Sをスタート地点として各頂点vに対してSからvにたどりつくまでに経由した頂点の数が多いパスを求めます。
これは頂点数が高々2000個なのでO(N^2)でも間に合います。
そのため、ベルマンフォード法と同じ要領で求められます。
これで最も長いパスが強連結分解後の集合の数と一致すればYesです。

```
vector<int> id(num,0);
id[group[1]] = 1;
int mx = 1;
for(int i=0; i<num; i++){
    for(int j=0; j<num; j++){
        if(!SCC[i][j]) continue;
        chmax(id[j],id[i]+1);
        chmax(mx,id[j]);
    }
}
```



実装例

実装は重めだと思います。

https://onlinejudge.u-aizu.ac.jp/services/review.html#ACPC2020Day3/4860637

Writer解/統計

Writer解
- rsk0315(C++ : 217行)
- monkukui(C++ : 135行)
- TAB(C++ : 135行)
- pitsu(C++ : 120行)

統計
- AC率　57/221
- FA　SSRS(18:44)