AtCoder Beginner Contest 021
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始める前に

AtCoder

- 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　**足し算**

1. 問題概要
2. アルゴリズム

A問題　問題概要

AtCoder

- 整数 N が与えられる
- 2の累乗数(1,2,4,8)のみの和で表わせ.
- 同じものを使ってもよい.
- 制約
  - 1 ≦ N≦ 10

A問題　アルゴリズム

AtCoder

- 同じものを使ってもよいという制約があるので1だけ使って表せばよい
- 組み合わせを構成する個数としてNを出力した後，N個の1を出力すれば満点

入力例

4

このアルゴリズムでの出力例

4

1

1

1

1

A問題　アルゴリズム

AtCoder

- 別解
  - 同じものを使ってもよいという制約はなくても解ける
  - Nの2進表現を考えれば，異なる2の累乗数(1,2,4,8,16,…)を使って任意のNが表せる
  - Nの2進表現において下からk桁目のビットが立っているなら2^kを組み合わせに加える.
  - ビット演算についてはネットで検索しましょう

B問題　嘘つきの高橋くん

1. 問題概要
2. 考察
3. アルゴリズム
4. おわりに

B問題　問題概要

AtCoder

- 町がN個ある. いくつかの町が道路によって結ばれている. 道路は双方向移動可能
- 道路の構成は全く分からない
- 高橋くんは町 a から出発して町 b に到着した
- 途中で経由した町の番号が全て分かっている
- 高橋くんが最短経路でaからbに移動した可能性はある？

- 制約
- 2≦N≦100

B問題　考察

AtCoder

- 以下の場合は，絶対に最短経路になりえない
  - 同じ町を2度以上経由した
  - 始点と同じ町を1度でも経由した
  - 終点と同じ町を1度でも経由した
- なぜなら，入力例にもあるように必ず同じ町から同じ町への移動は短絡できるから
- 逆にそれ以外の場合は，最短経路になりえる

(適当な直線グラフを考えるだけで明らかに最短経路になる)

- 結論として{始点,終点,経由した町の列}に重複する要素がなければよい.

B問題　アルゴリズム

AtCoder

- 長さnの配列に同じ要素が含まれるか判定する方法
- アルゴリズム1
  - 二重ループで判定
  - O(n^2)
- アルゴリズム2
  - 配列をソートした物の隣接する要素に同じものがないか判定
  - O(n log n)
- アルゴリズム3
  - 要素の最小値，最大値が存在する場合(今回，1≦(要素)≦N) は，出現頻度を数える配列を用意して数える
  - 出現頻度2以上の町が2つ以上あったらダメ
  - 値の範囲の大きさを C として，O(C+N)
- どれでもよい

B問題　おわりに

AtCoder

- 今回得られる知見
  - 負の重みがないグラフ(無向でも有向でも)においては
    - ある最短経路に現れる頂点は全て異なる
    - 最短経路は閉路を持たない

C問題　正直者の高橋くん

1. 問題概要
2. おことわり
3. 考察
4. アルゴリズム

C問題　問題概要

AtCoder

- N個の町とM個の道路がある.
- 町aから町bへの最短経路の個数を出力せよ
- 道路は双方向移動可能

- 制約
  - $2 \leqq N \leqq 100$
  - $1 \leqq M \leqq 200$
  - 最短経路とは辿る道路の本数が最小の経路と定義

C問題　おことわり

AtCoder

- この問題は，N個の頂点とM個の重み1の無向辺が与えられる場合のグラフの問題です．
- 以降そのような問題として説明．

C問題　考察

AtCoder

- まず，頂点aから全ての頂点への最短距離を求める
- それらの最短距離を$S_1,S_2,\ldots,S_N$としておく

- 道路が結んでいる辺を(i,j)とするとき，

$$S_i + (\text{辺のコスト，今回は1}) = S_j$$

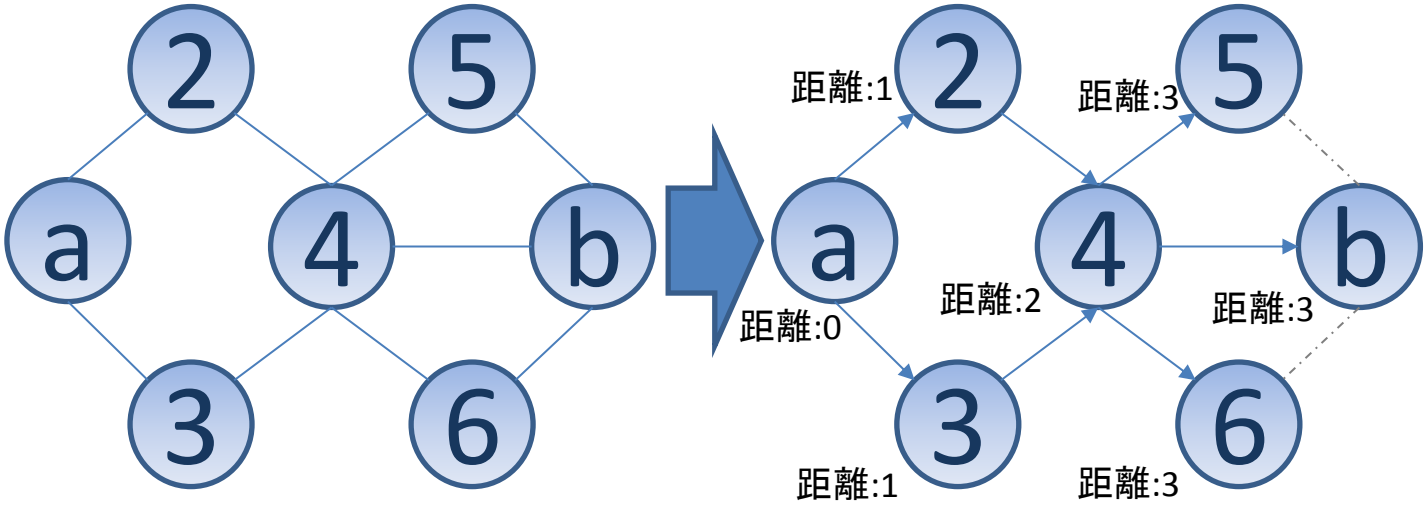
- が成り立つような全ての辺(*)で有向グラフ(最短路DAG)を作れば，そのグラフをどのように辿っても絶対に最短路を達成可能.

(*)双方向に移動可能なので，(x,y)という辺が与えられたら(y,x)という辺も与えられたものとする.

C問題　考察

AtCoder

- 最短路DAGの性質
  - DAG(Directed Acyclic Graph) なので，閉路がない有向グラフ
- グラフ(左)に対するaを始点とする最短路DAG(右)



- このDAG上で，aから b にたどり着く経路の個数を数える動的計画法を行えば，最短経路の答えとなる.

C問題　アルゴリズム

AtCoder

- 「頂点aから全ての頂点への最短経路DAGを構築」
  - ワーシャルフロイド法
  - 幅優先探索
  - ダイクストラ法

を行ってから，先ほどの条件を満たす辺のみグラフを構築

- 「DAG上の経路を求める」
  - トポロジカルソートを行ってから経路数え上げ動的計画法
  - そんなのはよくわからないという人は，始点aからメモ化再帰をすると意識しなくてよい
  - 動的計画法の計算量は O(N+M)

D問題　多重ループ

1. 問題概要
2. 考察
3. アルゴリズム
4. あとがき

D問題　問題概要

AtCoder

- nとkが与えられる.
- 以下のような多重ループにおける最終的なansの値を10^9+7で割った余りを出力せよ

```
ans←0
for a_1=1..n
  for a_2=a_1..n
    for a_3=a_2..n
      …
      for a_k=a_(k-1)..n
        ans ← ans+1
```

- 制約
  - 1≦N≦100,000
  - 1≦K≦100,000
  - 部分点(99点)では 1≦N≦1000,1≦K≦1000

D問題　考察

AtCoder

- $1 \leqq a_1 \leqq a_2 \leqq \ldots \leqq a_k \leqq n$
- であるような(a1,a2,…,ak)の組を数えれば良いというヒントがある.
- n個の中からk個を選ぶ組み合わせの数を$_nC_k$と表し,以降説明していく.

D問題　考察

AtCoder

- 直接の解法にはならないが少し問題を簡単化して，

$$1 \leqq a_1 < a_2 < \ldots < a_k < n$$

- が条件だとしたらどうだろう．
- 1からNの中からk個の数を選び，それらが全て異なる数であれば，こうなるように並べることができる．
- したがって，1からNの中からK個の数を選ぶ組み合わせに対応．
- この問題であればnCkを計算すればそれが答え

D問題　考察

AtCoder

- 元の条件に戻って考える．元の条件は

$$1 \leqq a_1 \leqq a_2 \leqq \ldots \leqq a_k \leqq n$$

- であった．
- 1からNの中から重複を許してk個の数を選び，重複を許して全て異なる数であれば，こうなるように並べることができる．
- したがって，1からNの中から重複を許してK個の数を選ぶ組み合わせに対応．
- このような組み合わせは「重複組合せ」と呼ばれ，${}_nH_k$と表される．

D問題　考察

AtCoder

$$ {}_nH_k = {}_{n-1+k}C_{n-1} $$

- という公式があります．これを計算すれば元の問題の答えです．
- なぜそうなるかはよく理解しておいたほうが良いでしょう．
- 各自ネットで調べると分かりやすい説明がいくつか出てくるので，重複組合せを知らなかった人は是非この機会に理解するとよいでしょう．

D問題　アルゴリズム

AtCoder

- nCrを計算する方法1
  - ${}_nC_0 = 1$, ${}_nC_r = {}_{n-1}C_{r-1} + {}_{n-1}C_r$
  - という公式を用いて，動的計画法を行う．余りは逐次取る
  - パスカルの三角形を構築すると言い換えてもよい
  - 計算量は O(n^2)
- これだと元の制約(1≦N≦10^5,1≦K≦10^5)では間に合わない

- nCrを計算する方法2
  - P=10^9+7(素数)で割るということに着目
  - 1!～N! mod P の逆元が存在
  - 1!,2!,…,N! mod P とそれらの逆元を全て予め計算しておく.
  - その上で，${}_nC_r = \frac{n!}{r!(n-r)!}$という公式を用いればO(1)
  - 前処理は O(N log P) (ちょっと改善すれば O(N+logP))

D問題　アルゴリズム

AtCoder

- 逆元テーブルの求め方
  - P が素数のとき，x(Pの倍数でない)の逆元x^{-1}は

$$x^{-1} = x^{p-2} \bmod P$$

  である.
  - フェルマーの小定理と呼ばれる
  - 二分累乗法を用いれば1つの逆元はO(log P)で計算可
  - だから，1!,2!,…,N!の逆元テーブルは，O(N log P)で生成可
  - 最初だけN!の逆元$\frac{1}{N!}$をO(log P)で計算し，次に

$$\frac{1}{(N-1)!} = \frac{1}{N!} \times N$$

  - であることを利用して(N-1)!を計算,…というふうに
    逆から逐次求めると，テーブルをO(N+logP)で作れる

D問題　あとがき

AtCoder

- 正直のところ，OEIS(オンライン数列大辞典)を用いて実験を行うと，よくわからなくても規則性が見えてくることがあります．
- オンライン数列大辞典のURL https://oeis.org/