

天下一プログラマーコンテスト 2016 予選 A 解説

問題一覧： <https://tenka1-2016-quala.contest.atcoder.jp/assignments>

A 天下一プログラマーゲーム

答え

2632 です。

解説

天下一プログラマーゲームでそのまま発言する整数は 3 でも 5 でも割り切れない数です。

なので、1 から 100 までの整数のうち、3 でも 5 でも割り切れない数をすべて足すと答えが得られます。

別解として、

$(100 \text{ までの総和}) - (100 \text{ までの } 3 \text{ の倍数の総和}) - (100 \text{ までの } 5 \text{ の倍数の総和}) + (100 \text{ までの } 15 \text{ の倍数の総和})$

を計算しても解を求めることができます。

B PackDrop

解説

この問題では、ネットワークは根付き木です。

2つの頂点間にはさんである PackDrop の個数を、2 頂点間の辺の重みと呼ぶことにします。また、根から頂点 v までの経路上にある PackDrop の個数を、頂点 v の重みと呼ぶことにします。

すると、この問題は、根付き木のそれぞれの葉の重みが与えられたとき、辺の重みの総和を最小化する問題であると言い換えることができます。

ある頂点 v について、 v の親とを結ぶ辺の重みを 1 増やしても、 v の全ての子を結ぶ辺たちの重みを 1 増やしても、葉の重みは変わりません。したがって、できるだけ根に近い頂点の重みを大きくとることで、辺の重みの総和を小さくできます。

余談

PackDrop は KLab で実際に使用されているネットワークエミュレーターで、遅延やパケットの欠落をエミュレートすることができます。不安定なネットワークで起きるバグの再現・修正に役立っています。

DSAS 開発者の部屋:ラズパイで作るネットワークエミュレータ (前編)

<http://dsas.blog.klab.org/archives/raspi-netem1.html>

DSAS 開発者の部屋:ラズパイで作るネットワークエミュレータ (後編)

<http://dsas.blog.klab.org/archives/raspi-netem2.html>

C 山田山本問題

解説

a から z までのアルファベットをトポロジカルソートする問題です。

入力のサイズは大きいのですが、実はそれぞれのアカウント名の組に対しインデックス順で最初に異なった文字だけを気にすれば良いことになります。

DFS を用いたアルゴリズムも広く知られていますが、この問題の場合は辞書順最小を選択する必要があるため、入次数 0 の頂点から探索する類のアルゴリズム (Kahn's algorithm) を用いると簡単に解けるでしょう。

D グラフィカルグラフ

解説

出力できるのは 100×100 マスまでという制約がありますが、まずはこの制約を外して、どれだけ広い領域を使ってもよい（メモリに収まらないほど広くてもよい）として考えてみましょう。

その場合の解法の一例を示します。

適当な頂点を根とし、適当な位置に置きます。

その頂点と隣接する頂点を、根から上下左右に 2^N マス離れたところに置きます。

さらに、いま置いた頂点と隣接する頂点を、上下左右に $2^{(N-1)}$ マス離して置きます。

このように頂点を置いていけば、重なることなくすべての頂点を配置できます。

しかし、元の問題で使えるのは 100×100 マスまでです。

どうすればよいでしょうか？

実は単純で、マスの数の制約を取り払って得た解で、

頂点のない行が二つ以上連続しているなら、それらの行を圧縮して

一つの行にまとめることができます。

同様に列も圧縮することで、たかだか $2N \times 2N$ ($2N = 52$) マスの解が得られます。

E 無限グラフ

解説

入力から、次のような N 頂点 $2M$ 辺の有向グラフを作ります。

各頂点は $0 \sim N-1$ の番号を持ち、入力の各組 (A_i, B_i) に対し、

A_i から B_i へのコスト 1 の辺と、

B_i から A_i へのコスト -1 の辺を持ちます。

このグラフの連結成分ごとに、存在するすべてのサイクルのコストの最大公約数を足し合わせたものが答えとなります。

(サイクルが存在しない連結成分、またはコスト 0 のサイクルしか存在しない連結成分が存在する場合、答えは -1 となります)

その値を計算するためには、このグラフのサイクル基底をすべて列挙し、

それらのサイクルのコストの最大公約数を求めればよいです。

これは、一つの頂点から単純に DFS を行うことで計算できます。

なお、部分点だけを得るには、グラフ全体から適当な長さの区間を取り、

その区間で連結成分の個数を直接求めればよいです。