



@drken 2020年01月30日に更新



AtCoder 版！蟻本（中級編）

アルゴリズム

AtCoder

競技プログラミング

新人プログラマ応援

蟻本

0 はじめに

前回の[初級編](#)に続いて、今度は中級編です。

[プログラミングコンテストチャレンジブック](#) (通称、蟻本) は日本の競技プログラミングの普及に多大な貢献を果たしています。多くの競技プログラマたちが蟻本を手に取りながらコンテストの世界に没入して行きます。しかしながら発売から 6 年以上経過する間に競技プログラミング界隈には大きな変化がありました。蟻本的に影響が大きいのは以下の点です:

- [POJ](#) が国内ではあまり使用されなくなった (計算速度が遅いなど)
- [AtCoder](#) 上で問題を解くことが盛んになった

今回はこの完全解決を試みます。具体的には、蟻本に載っている例題たち (ほとんどすべて POJ 上の問題です) を AtCoder 上でジャッジできる問題に対応付けようという試みです。今回は中級編を扱います。AtCoder 上で見つからなかったものは [AOJ](#), [yukicoder](#) 上の問題も載せていきます。

【注意】

- 本記事の趣旨からしてどうしてもネタバレ問題は生じてしまいますので、それについては了承いただければと思います。
- 今後随時更新していきますので、各問題の解説などの参考記事やよりよい問題案などの意見をガンガン募集しています。
- 関連プロジェクトとして、こうきさんによる Atcoder 過去問に自動的にタグをつけるサービスの [AtCoder Finder](#) や、類似プロジェクトの [Project Dijkstra](#)、つたじゅー☆さんによる

蟻本練習問題解説があります。皆で相補的に活動できたらいいなと思います。

【シリーズ】

- [AtCoder に登録したら次にやること ～ これだけ解けば十分闘える！過去問精選 10 問 ～](#)
- [AtCoder 版！蟻本 \(初級編\)](#)
- [AtCoder 版！蟻本 \(上級編\)](#)
- [AtCoder 版！蟻本 \(発展的トピック編\)](#)

3 ここで差がつく！ --- 中級編

3-1 値の検索だけじゃない！ "二分探索"

例題 3-1-1 lower_bound

【キーワード】

- 二分探索
- lower_bound

【AtCoder 上の類題】

- [ABC 077 C Snuke Festival](#) (lower_bound を試すのによい問題です)
- [ARC 037 C 億マス計算](#) (k 番目を求めるのにも二分探索を適用できます)
- [JOI 2008 本選 B ピザ](#) ([AOJ 0539](#) に同じ)
- [JOI 2007 本選 C ダーツ](#) ([AOJ 0529](#) に同じ。初級編の一番最初の問題 (くじびき) とほぼ同じ問題です)

【コメント】

lower_bound と upper_bound、よく頭がこんがらがります> <

例題 3-1-2 Cable Master (POJ No.1064)

【キーワード】

- 二分探索
- 解を仮定し可能か判定 (判定問題にするテク)
- 連続値に対する二分探索
- 方程式を解く二分探索

【AtCoder 上の類題】

- [ABC 023 D 射撃王](#) (よい例題です)
- [ARC 050 B 花束](#) (よい例題です)
- [ABC 026 D 高橋君ボール1号](#) (連続量に対する二分探索... どちらかと言うと二分法です)
- [AOJ 2220 The Number of the Real Roots of a Cubic Equation](#) (方程式を解きます)

【コメント】

「～を満たす最大 (最小) の値を求めよ」という問題に対して、

- 解 x を仮定して可能かを判定する問題に置き換えて
- 解 x が可能となるような最大 (最小) の x を求める

というのはあまりにも頻繁に出て来るテクニックです！

x という解がとりあえず存在すると仮定することによって、Greedy に判定問題を解くことができるようになるケースが多いです。

また Cable Master は連続値に関する二分探索でもあるので、その筋の問題も加えました。

例題 3-1-3 Aggressive Cows (POJ No.2456)

【キーワード】

- 二分探索
- 最小値の最大化 (最大値の最小化)

【AtCoder 上の類題】

- [CODE FESTIVAL 2015 予選A D 壊れた電車](#) (いい感じの例題です、くぬーさん提供)
- [JOI 2013 本選 C バームクーヘン](#) (しゃくとり法の要素もあって少し難しいです、[AOJ 0600](#) に同じ)
- [JOI 2016 本選 C JOIOI 王国](#) (判定問題に落としてからの Greedy パートが難しめです。実は二分探索しなくても解けます。)

【コメント】

最小値の最大化を見たら条件反射で二分探索したくなります。なぜなら

$$\min(f(i)) \geq K \Leftrightarrow \text{すべての } i \text{ に対して } f(i) \geq K$$

という関係が成り立つからです。いかにも二分探索のフレームワークがはまりそうですね。

判定問題に落とした後は Greedy であることが多いです。

Greedy パートの難易度は問題によって大きく変わって来ます。

例題 3-1-4 平均最大化

【キーワード】

- 二分探索
- 平均値の最大化
- 「 $a[0], \dots, a[n-1]$ の**平均**が K 以上」を「 $a[0]-K, \dots, a[n-1]-K$ の**総和**が 0 以上」にするテク

【AtCoder 上の類題】

- [ABC 034 D 食塩水](#) (そのもの)
- [ARC 026 D 道を直すお仕事](#) (少し難しめです)
- [ARC 060 C 高橋君とカード](#) (二分探索ではないですが**平均を総和にするテク**を使うとオーダーが落ちます)
- [ARC 075 E Meaningful Mean](#) (やはり平均を総和にするテク、下に出て来る**転倒数**も絡みます)

【コメント】

「 $a[0], \dots, a[n-1]$ の平均が K 以上」を

「 $a[0]-K, \dots, a[n-1]-K$ の総和が 0 以上」にするテクは
二分探索に限らずちょくちょく見る印象です。

単純な二分探索の問題は AtCoder 上だと意外と少ないですね。
はまやんさんのまとめに、より色々な問題があります

競技プログラミングにおける二分探索・三分探索問題まとめ

3-2 厳選！頻出テクニック (1)

例題 3-2-1 Subsequence (POJ No.3061)

【キーワード】

- しゃくとり法

【AtCoder 上の類題】

- AOJ Course The Number of Windows (しゃくとり法の練習に)
- ABC 032 C 列 (ほぼそのまんまです)
- ABC 038 C 単調増加 (しゃくとり法は最大区間・最小区間を求めるだけでなく数え上げにも有効です)
- ARC 098 D Xor Sum 2 (同じく数え上げにも有効です)

【コメント】

高難易度問題において部分的に登場するイメージが強いテクです。

求めるものが「最小の区間」か「最大の区間」かはあまり影響がないです。POJ の例題は最小区間ですが、ABC 032 C は最大区間です。そればかりか、しゃくとり法は条件を満たすものを
列挙するタイプのアルゴリズムなので数え上げもできます。

例題 3-2-2 Jessica's Reading Problem (POJ No.3320)

【キーワード】

- シャクとり法
- 前処理

【AtCoder 上の類題】

- [ARC 022 B 細長いお菓子](#) (例題と非常によく似ています)
- [ABC 017 D サプリメント](#) (DP + シャクとり法)
- [ABC 033 D 三角形の分類](#) (角度ソートとの組み合わせ、少し面倒です)
- [JOI 2013 本選 C バームクーヘン](#) (**二分探索**との組み合わせ)
- [JOI 2017 予選 E L番目のK番目の数](#) (ついこの間の JOI 予選最終問です)

【コメント】

例題 3-2-1 と大体一緒です。例題 3-2-1 で挙げたものに比べると他分野との複合的な問題になっているものを載せました。

- [競技プログラミングにおける尺取法問題まとめ](#) (はまやんさん)
-
-

例題 3-2-3 Face The Right Way (POJ No.3276)

【キーワード】

- 反転 (ライツアウト)
- XOR 操作
- 2 回やったら元に戻る
- 操作の順序を入れ替えても結果は変わらない
- 端から順に決まる Greedy
- 差分更新

【AtCoder 上の類題】

- [ABC 048 C Boxes and Candies](#) (反転ではないですが、端から決まっていく感じは似ています)
- [ABC 083 D Wide Flip](#) (一見似ていますが全然違う問題です)
- [JOI 2012 本選 A 電飾](#) (それほど似ていないですが面白いです, [AOJ 0603](#) に同じ)
- [JOI 2008 本選 C あみだくじ](#) (差分更新系の問題も挙げます, [AOJ 0540](#) に同じ。)
- [RUPC 2018 day1-C 一致](#) (差分更新していく感じはよく似ています)
- [KUPC 2012 H 植林](#) (少し難しめです。2 回やったら元に戻る性質をフルに考察します)

【コメント】

ライツアウト系の問題 (決まった領域の反転を繰り返して全部白にしたりする系) です。

例題 [POJ No.3276](#) の難易度は高めですが非常に学ぶ要素の多い問題です。

XOR 操作においてしばしば以下のような考察が重要になります。

- 2 回やったら元に戻る (逆元が自分自身)
- 操作の順序を入れ替えても結果は変わらない

この問題の場合、さらに「**端から考える**」ことによって次々と Greedy に決まっていきます。さらに毎ステップごとの操作を実施するのが計算量的に厳しくても「**差分を更新すればいい**」という発想も頻出です。

例題 3-2-4 [Fliptile \(POJ No.3279\)](#)

【キーワード】

- 反転 (ライツアウト)
- XOR 操作
- 操作の順序を入れ替えても結果は変わらない
- なにかを決めると Greedy

【AtCoder 上の類題】

- [JOI 2007 予選 E おせんべい](#) (比較的近い感じですが, [AOJ 0525](#) に同じ)
- [ABC 018 D バレンタインデー](#) (見た目は違いますが、なにかを決めると残りが Greedy に決まる感じは似ています)

- [yukicoder No.226 0-1パズル](#) (例題にかなり近いと思います)
- [AOJ 1308 Awkward Lights](#) (ライツアウトです、**連立一次方程式**を解きます)

【コメント】

ライツアウト第二弾です！

今度は端から考えようとしてもいきなり一意には決まりません。

しかし最初の一行を全部決めると、それより下は全部 Greedy に決まっています。

このように「なにかを決めると残りは Greedy に決まる」というのも頻出の印象です。

また、Greedy 系の解法ではいかんともしがたいライツアウト系に対しては、

連立一次方程式を解く解法も検討するとよさそうです。

例題 3-2-5 [Physics Experiment \(POJ No.3684\)](#)

【キーワード】

- 弾性衝突
- 物理

【AtCoder 上の類題】

- [AGC 013 C Ants on a Circle](#) (例題 1-6-2 でも挙げた、蟻のオマージュにして類似の発想をする問題です...その段階では厳しくても中級編を学ぶ今の段階では、ちょうど解き時と言えそうです)
- [SRM 458 DIV1 Easy BouncingBalls](#) (SRM の問題ですが、期待値の線形性も同時に学べる良問です、是非 AtCoder にも欲しい...)

【コメント】

AtCoder では物理系の問題はあまり見かけないかもです。

例題 3-2-6 [4 Values whose Sum is 0 \(POJ No.2785\)](#)

【キーワード】

- 半分全列挙
- 二分探索

【AtCoder 上の類題】

- [JOI 2007 本選 C ダーツ](#) ([AOJ 0529](#) に同じ)

【コメント】

初級編の一番最初の問題「くじびき」と大体一緒ですね。

これも初級編の最初の段階では厳しくても、今なら解き頃と言えそうです。

例題 3-2-7 巨大ナップサック

【キーワード】

- 半分全列挙

【AtCoder 上の類題】

- [AOJ Course Huge Knapsack Problem](#) (手法の verify に)
- [ABC 032 D ナップサック問題](#) (例題 2-3-4 のテクも使います)
- [AGC 026 C - String Coloring](#) (半分全列挙と見抜けさえすればという感じです)
- [ARC 017 C 無駄なものが嫌いな人](#)
- [CODE THANKS FESTIVAL 2017 G Mixture Drug](#) (いわゆる**最大安定集合問題**です)

【コメント】

例題 3-2-6 は $O(n^4)$ を $O(n^2 \log n)$ にする系の半分全列挙でしたが、今度は $O(2^n)$ を $O(n \cdot 2^{(n/2)})$ にする系の半分全列挙です。

最後の問題は「一般グラフの最大安定集合問題」そのものですが、 $n \leq 40$ 程度なら半分全列挙で解けます。さらに計算時間を落とした $O(1.381^n)$ のシンプルなアルゴリズムもあります:

参考記事:

- [指数時間アルゴリズム入門](#) (wata さん)
 - [JOI 2015 予選6：財宝の、想定解法よりも効率的な解法の解説](#) (square さん)
 - [競技プログラミングにおける半分全列挙問題まとめ](#) (はまやんさん)
-
-

例題 3-2-8 領域の個数

【キーワード】

- 座標圧縮

【AtCoder 上の類題】

- [JOI 2007 本選 E ペンキの色](#) (かなり近い問題です、[AOJ 0531](#) に同じ)
- [JOI 2012 予選 E 魚の生息範囲](#) (今度は三次元です！[AOJ 0580](#) に同じ)
- [JOI 2008 予選 E シャッフル](#) (領域に関する問題ではないですがこれも立派な座圧と言えます、[AOJ 0536](#) に同じ)
- [ABC 008 D 金塊ゲーム](#) (区間 DP との組み合わせです、ABC D 問題としては難しいです)
- [APC 001 I Simple APSP Problem](#) (難しいですが、僕が今までに解いた座圧で一番感動的かもしれません)

【コメント】

出ました！！！！あまりにもよく使用するテクニック、座圧です！！！！

3-3 さまざまなデータ構造を操ろう

例題 3-3-1 [Crane \(POJ No.2991\)](#)

【キーワード】

- セグメントツリー

【AtCoder 上の類題】

- [AOJ Course Range Minimum Query \(RMQ\)](#) (とりあえず RMQ や、初めて書く segtree の verify に)
- [JOI 2011 春合宿 day4-2 本棚](#) (RMQ を用います。問題は[ここ](#))
- [Japan Alumni Group Summer Camp 2013 Day 2 G Perm Query](#) (面白い感じです)
- [ARC 008 D タコヤキオイシクナール](#) (セグメントツリーの演算は交換法則を満たさなくてもいいです)
- [みんなのプロコン 2017 D 工場](#)

【コメント】

特に遅延評価などを伴わないセグメントツリーの問題です。タコヤキオイシクナールは、セグメントツリーの演算が交換法則を満たさなくてもいいことを映し出す教育的問題と言えそうです。

参考記事:

- [データ構造と代数構造](#) (koba さん)
-
-

例題 3-3-2 バブルソートの交換回数

【キーワード】

- BIT
- 転倒数

【AtCoder 上の類題】

- [Chokudai SpeedRun 001 J- 転倒数](#)
- [AOJ Course The Number of Inversions](#) (バブルソートの交換回数そのものです)
- [Indeedなう 2015 E Line up!](#) (重みつき転倒数という感じです)
- [Tenka1 2017 E CARtesian Coodinate](#) (転倒数)
- [ARC 088 E Papple Sort](#) (転倒数)

- [ARC 033 C データ構造](#) (BIT + 二分探索が自然ですが、平方分割などの解法も考えられます)
- [第2回早稲田大学プログラミングコンテスト G だるま落とし](#) (BIT + 二分探索)
- [JOI 2010 本選 E 微生物実験](#) ([AOJ 0564](#) に同じ)

【コメント】

ここんところ AtCoder 上でも頻出の転倒数 (BIT や分割統治法で求めます) です。

BIT 上二分探索を用いるタイプの問題もここに載せました。BIT については hos さんの神資料があります:

- [Binary Indexed Tree のはなし](#)
-
-

例題 3-3-3 [A Simple Problem with Integers \(POJ No.3468\)](#)

【キーワード】

- 区間加算
- 遅延評価つきセグメントツリー
- 区間加算対応 BIT

【AtCoder 上の類題】

- [AOJ Course RSQ and RAQ](#) (まったく同じです)
- [AOJ Course RMQ and RAQ](#) (「区間加算」と「最小値」、いわゆる有名な **Starry Sky Tree** です!)
- [JOI 2016 春合宿 day3-2 雇用計画](#) (座標圧縮もします、問題は[ここ](#))
- [CODE FESTIVAL 2015 決勝 D 足ゲームII](#) (いもす法で解けますが、Starry Sky Tree があれば貼るだけです)
- [square869120Contest #2 H Counting 1's](#) (Starry Sky 以外のを)
- [JOI 2012 春合宿 day3-2 かくれんぼ](#) (問題は[ここ](#))
- [ARC 076 F Exhausted?](#) (面白いです)

【コメント】

いわゆる遅延評価つきセグメントツリーです。遅延評価セグメントツリーについてはよい資料と

問題集がたくさんあるので以下に貼ります:

- [僕のセグメントツリーの使い方](#) (きゅうりさん)
 - [遅延評価セグメント木をソラで書きたいあなたに](#) (つたじろうさん)
 - [データ構造と代数\(後編\)](#) (tomcatowlさん)
 - [初心者の初心者による初心者のための典型segment tree](#) (DEGwerさん)
 - [競技プログラミングにおけるセグメントツリー問題まとめ](#) [セグメントツリー](#), [BIT](#), [遅延評価セグメントツリー](#) (はまやんさん)
-
-

例題 3-3-4 [K-th Number \(POJ No.2104\)](#)

【キーワード】

- 平方分割

【AtCoder 上の類題】

- [ARC 033 C データ構造](#) (例題 3-3-2 のところでも挙げましたが平方分割でも解けます)
- [ARC 042 D あまり](#) (離散対数を求める **Baby-Step Giant-Step** を用いますが、これも平方分割の一種と言えます)
- [KUPC 2014 K 乱数調整](#) (同じく BSGS です)
- [JOI 春合宿 2016 day3-2 回転寿司](#) (難しいです、問題は[ここ](#))
- [第3回 ドワンゴからの挑戦状 本選 B ニワンゴくんの約数](#) (最近話題の **Mo のアルゴリズム** で解けますが、Mo のアルゴリズムも平方分割の一種と言えます)

【コメント】

平方分割は極めて汎用性の高いテクニックですね。平方分割以外が想定解法でも平方分割で解ける問題は非常に多いです。例題 3-3-1~3 の類題で挙げた問題たちも、平方分割でも解けるものは多いです。

参考記事:

- [セグメント木をあきらめた人のための平方分割](#) (しょラーさん)
- [競技プログラミングにおける平方分割問題まとめ](#) (はまやんさん)
- [競技プログラミングにおけるMo's Algorithm問題まとめ](#) (はまやんさん)

3-4 動的計画法を極める！

例題 3-4-1 巡回セールスマン問題

【キーワード】

- bit DP
- TSP

【AtCoder 上の類題】

- [JOI 2016 予選 D めいぐるみの整理](#) (とてもよい bit DP の例題です)
- [AOJ Course Traveling Salesman Problem](#) (巡回セールスマン問題です)
- [ABC 041 D 徒競走](#) (トポロジカルソートの数え上げです。bit DP の例題としてはとてもよいです。トポソ数え上げは一般グラフでは #P-complete ですが、ツリー上ではツリー DP で効率よく解けます)
- [JAG Practice Contest for ACM-ICPC Asia Regional 2013 C SIRO Challenge](#) (重ための TSP です)

【コメント】

ついに bit DP まで来ました。ナイーブに考えると $n!$ 通り探索する必要のありそうな問題に対して、 2^n 通り程度の状態考えればいようにする使い方が最もよく見られます。

例題 3-4-2 [Traveling by Stagecoach \(POJ No.2686\)](#)

【キーワード】

- bit DP

【AtCoder 上の類題】

- [みんなのプロコン 2018 C 駆引取引](#) (ゲームっぽさが絡みます)
- [ABC 025 D 25個の整数](#)
- [ARC 016 C ソーシャルゲーム](#) (期待値との絡みです)
- [TDPC J ボール](#) (典型的な bit DP ですが多少式変形が必要です)

【コメント】

例題 3-4-2 は、例題 3-4-1 と大体一緒です。類題としては、例題 3-4-1 で挙げたものに比べると他分野との複合的な問題になっているものを載せました。

例題 3-4-3 ドミノ敷き詰め

【キーワード】

- bit DP
- bit をスライドさせる感じの bit DP

【AtCoder 上の類題】

- [JOI 2010 予選 F JOI旗](#) (比較的似た問題です、[AOJ 0559](#) に同じ)
- [CODE FESTIVAL 2014 予選A D 壊れた電卓](#) (桁 DP との組み合わせです)
- [TDPC Q 連結](#) (ナップサック DP を進めていくにつれて bit 状態がスライドしていくタイプの DP です、個人的に **sliding bit DP** と呼んでいます)
- [ARC 058 E 和風いろはちゃん](#) (同じく sliding bit DP です)
- [JOI 2012 予選 F お土産購入計画](#) (遷移を詰めるのがちょっと大変です)

【コメント】

ナップサック DP と組み合わせさった感じの bit DP を集めました。

参考記事:

- [競技プログラミングにおけるbitDP問題まとめ](#)
-
-

例題 3-4-4 フィボナッチ数列

【キーワード】

- 行列累乗
- m 項間漸化式を行列累乗に持ち込む

【AtCoder 上の類題】

- [ABC 009 D 漸化式](#) (演算が特殊ですが行列累乗で解けます)
- [ARC 050 C LCM 111](#)
- [SRM 428 DIV1 Medium TheLongPalindrome](#)
- [TDPC T フィボナッチ](#) (これは行列累乗でも間に合わず、きたまさ法です)

【コメント】

決まると気持ちのいい技、行列累乗です！

ここでは数式変形を頑張って行列累乗に持ち込む系の問題を挙げました。

例題 3-4-5 Blocks (POJ No.3734)

【キーワード】

- DP
- 行列累乗

【AtCoder 上の類題】

- [TDPC M 家](#) (bit DP との組み合わせで、最後は行列累乗になります)
- [AGC 006 C Rabbit Exercise](#) (適度な感じです)
- [AGC 013 E Placing Squares](#) (DP をとにかく頑張って立てて行列累乗に持ち込みます)

【コメント】

dp を立てて行列累乗に持ち込むタイプの問題です。

例題 3-4-6 グラフの長さ k のパスの総数

【キーワード】

- 行列累乗

【AtCoder 上の類題】

- [Educational DP Contest R - Walk](#) (まさにグラフの長さ k のパスの総数を求める問題です)
- [Japan Alumni Group Summer Camp 2013 Day 4 F Graph Automata Player](#) (英語ですがいい感じの問題です。最後は連立一次方程式を解きます)
- [CSA IATI Shumen 2017 Day 1 B Superstition](#) (行列級数和的な処理も含みます、E869120 さん・square10011 さん提供)
- [SRM 342 DIV1 Hard DrivingAround](#) (当時は DIV1 Hard でも今なら Medium でも易しめな感じですよ)
- [SRM 405 DIV1 Medium AllCycleLengths](#) (SRM の問題ばかり載せてアレですが、この時代の SRM は行列累乗が流行っていました)
- [SRM 608 DIV1 Medium BigO](#) (最終的な解法は行列累乗ではないですが面白いです)

【コメント】

DP の様式としては例題 3-4-5 となんら変わらないですが、特にグラフの遷移行列 (やその亜種) について累乗していくタイプの問題です。

例題 3-4-7 [Matrix Power Series \(POJ No.3233\)](#)

【キーワード】

- 行列累乗
- 行列級数和

【AtCoder 上の類題】

- [SRM 446 DIV1 Medium AntOnGraph](#) (SRM の問題ですが面白いので載せます、単に行列累乗を繰り返すのでもできますが、行列級数和で一発で決めることもできます)

【コメント】

A^n を求めるのが行列累乗なら、 $E + A + A^2 + \dots + A^{(n-1)}$ を求めるのが行列級数和です。等比級数の和の公式のようなものを使いたくなるのですが、 $E - A$ が逆行列をもつとは限らないのでダメです。AtCoder 上の問題を[募集中](#)です。

例題 3-4-8 [Minimizing Maximizer \(POJ No.1769\)](#)

【キーワード】

- セグメントツリーによる DP 高速化
- セグメントツリー上の DP
- インライン DP (実家 DP)

【AtCoder 上の類題】

- [第4回 ドワンゴからの挑戦状 本戦 B だんだん強く](#)
- [JOI 2019 予選 E - イルミネーション \(Illumination\)](#)
- [ARC 073 F Many Moves](#) (sky さんの記事で解説されています)
- [ARC 085 F NRE](#) (区間を扱う系はインライン DP であることが多いようです)
- [JOI 2015 春合宿 day1-3 たのしいたのしい家庭菜園](#) (問題は[ここ](#))

【コメント】

最近 AtCoder 上でも流行っている[インライン DP](#) (sky さん) です。

行列累乗やインライン DP などについてはまやんさんの問題集:

[競技プログラミングにおける動的計画法更新最適化まとめ \(CHT, MongeDP, AlianDP, インラインDP, きたまさ法\)](#)

3-5 水を流して解く "ネットワークフロー"

例題 3-5-1 最大通信量

【キーワード】

- フロー
- 最大流
- 最大流最小カット定理

【AtCoder 上の類題】

- [AOJ Course Maximum Flow](#) (最大流問題の verify に)
- [yukicoder No.654 Air E869120](#) ([アルゴリズムデザイン](#)にも載っている典型的な最大流の適用例です)
- [ABC 010 D 浮気予防](#) (最小カット問題を解きたくなるので、最大流に帰着します)
- [ARC 074 F Lotus Leaves](#) (最小カットが自然に出て来るので、最大流に帰着します)
- [KUPC 2016 E 柵](#) (同じく)
- [JAG Practice Contest for ACM-ICPC Asia Regional 2014 F Reverse a Road II](#) (少し難しいですが、最大流最小カット定理の構造をきちんと理解しているかを問う良問です)

【コメント】

最大流問題です。問題設定からは最小カットの方が自然に思い浮かぶような問題 (最大流最小カット定理から最大流問題を解くことに帰着します) も含めました。

例題 3-5-2 仕事の割り当て

【キーワード】

- フロー
- 二部グラフ
- 二部グラフの最大マッチング

- 割り当て問題

【AtCoder 上の類題】

- [AOJ Course Bipartite Matching](#) (二部マッチングの verify に、問題のサイズは小さめ)
- [ABC 091 C 2D Plane 2N Points](#) (実は Greedy に解けるのですが、フローで殴れます)
- [AOJ 1163 カードゲーム](#) (とても典型的な二部マッチングです)
- [AOJ 2480 Blame Game](#) (難しめです、二部マッチングの**最適性条件**についての深い理解を問う良問です)
- [Indeed なう F 就職活動](#) (難しいです、結局フローを流すわけではないですが、**Hall の定理**を適用します)
- [ARC 076 F Exhausted?](#) (難しいです、セグメントツリーの類題としても挙げました。やはり Hall の定理を適用します)

【コメント】

最大流問題の最もよくある応用として、二部グラフの最大マッチングがあります。二部グラフの最大マッチングに関連する話題として **Hall の定理**があります。これは二部グラフに対する最大流最小カット定理や、**Dilworth の定理** (下に出て来ます) との等価性も知られており、ネットワークフロー理論の強力な定理群の 1 つの側面と考えることができます。

例題 3-5-3 二人組

【キーワード】

- 一般グラフの最大マッチング (not フロー)

【AtCoder 上の類題】

- [AOJ 2347 Sunny Graph](#) (Tutte 行列を少し変えて行列補完します、難しいです)
- [TOC 2006 Round1 Medium Separate Connections](#) (一般マッチングを verify する問題として長年君臨しています)
- [UOJ No.79 一般图最大匹配](#) (中国語ですが一般マッチングのようです)
- [ARC 080 F Prime Flip](#) (結局二部グラフの最大マッチングで解けるのですが、一瞬、一般マッチングが必要そうな気分になります)

- [RUPC 2018 day2 G Combine Two Elements](#) (結局二部グラフになるのですが、一般グラフの最大マッチングで殴れます)

【コメント】

一般グラフの最大マッチングです。大きく以下の 2 通りの方法が知られていますが、プログラミングコンテストでの出題歴は非常に少なく、もし「一般グラフの最大マッチング」に帰着されたと感じたときは違う解法があるケースが多そうです。

- Edmonds の花アルゴリズム
 - 行列補完 (Tutte 行列)
-
-

例題 3-5-4 最小コスト通信

【キーワード】

- フロー
- 最小費用流問題

【AtCoder 上の類題】

- [AOJ Course Minimum Cost Flow](#) (最小費用流の verify に)
- [AOJ 1088 School Excursion](#) (最小費用最大流です)
- [SRM 537 DIV1 Hard PrinceXDominoes](#) (最小費用循環流です)

【コメント】

最小費用流問題です

例題 3-5-5 [Asteroids \(POJ No.3041\)](#)

【キーワード】

- フロー
- 二部グラフ
- 二部グラフの最小点被覆・最大安定集合
- グリッド問題を二部グラフにする 2 つの方法

【AtCoder 上の類題】

- [SoundHound Inc. Programming Contest 2018 \(春\) C 広告](#) (パターン 2 の典型問題で最大安定集合問題です)
- [2015 天下一 予選A C 天下一美術館](#) (パターン 2)
- [yukicoder No.421 しろくろチョコレート](#) (パターン 2)
- [AOJ 2429 まるかいて](#) (パターン 1 で二部グラフにします、最小費用流になります)
- [Japan Alumni Group Spring Contest 2014 C Decoding Ancient Messages](#) (難しめですがパターン 1 の面白い問題です)

【コメント】

グリッドに関する問題はしばしば二部グラフにして考えることが有効ですが、2 つのパターンをよく見る気がします:

1. 左ノードが x 座標、右ノードが y 座標に対応して、各マス (x, y) は左右を結ぶ辺に対応する (Asteroids はこっち)
2. グリッドグラフは市松模様に塗ると二部グラフである

参考記事:

- [競技プログラミングにおける最大クリーク問題、最大独立集合問題まとめ](#) (はまやんさん)

例題 3-5-6 Evacuation (POJ No.3057)

【キーワード】

- フロー
- 最大流
- 二部マッチング (「人」と「(時刻, ドア)」との間の最大マッチング)
- 割当問題

【AtCoder 上の類題】

- [Maximum-Cup 2018 H Maxmin Tour](#) (二分探索して、「スタンプ間の移動」と「魔法の石」との間の最大マッチングです)
- [ARC 013 D 切り分けできるかな?](#) (条件設定が少し複雑です)
- [SRM 477 DIV1 Medium PythTriplets](#) (SRM ですが程よい感じの二部マッチングです)
- [SRM 594 DIV1 Medium FoxAndGo3](#) (マッチングというよりは最大安定集合です、「燃やす埋める」から考えることもできます)

【コメント】

いわゆる 2 つのカテゴリ間でマッチングを作るタイプの問題 (**割当問題**とも呼びます) です。類題は結構あると思ったのですが、多くの場合は重みもついていて (例題 3-5-10 で挙げます)、重みのないものは意外と見つからなかったです。

例題 3-5-7 [Dining \(POJ No.3281\)](#)

【キーワード】

- フロー
- 最大流
- 二部マッチングの亜種
- 割当問題の亜種

【AtCoder 上の類題】

- [UTPC 2012 J きたまさの逆襲](#) (最小費用流ですが、比較的近い発想を用いる問題です)

【コメント】

Dining は「食べ物」「飲み物」の両方を「牛」に割りあてるために単なる二部マッチングとは行かず、工夫が必要な例題でした。

例題 3-5-8 Dual Core CPU (POJ No.3469)

【キーワード】

- フロー
- 最小カット
- Project Selection Problem (通称「燃やす埋める」)

【AtCoder 上の類題】

- [ARC 085 E MUL](#) (PSP と呼ぶべきだという議論のきっかけになりました)
- [ARC 031 D 買い物上手](#) (少し難しいです)
- [AOJ 2903 Board](#) (とても面白いです)
- [KUPC 2017 H Make a Potion](#) (LP定式化して色々やると **2-SAT 制約**の問題となり、「燃やす埋める」が使える形式になります)

【コメント】

「燃やす埋める」の愛称で親しまれている Project Selection Problem です。よい資料が多数あるので以下に挙げます:

- 『燃やす埋める』と『ProjectSelectionProblem』 (とこはるさん)
 - 最小カットを使って「燃やす埋める問題」を解く (診断人さん)
 - 最小カットについて (yosupo さん)
 - 最小カットとProject Selection Problemのまとめ (うさぎ小屋さん)
-
-

例題 3-5-9 Farm Tour (POJ No.2135)

【キーワード】

- フロー
- 最大流・最小費用流
- 点素パス・辺素パスのパッキング

【AtCoder 上の類題】

- [AOJ 3047 Shiritori](#) (辺素パスの考え方になじんでいると簡単です)
- [TDPC R グラフ](#) (想定解法はもちろん DP ですが、フローでも解けることで有名です。結局 2 本の点素パスを流す問題に落ちます。)
- [AOJ 2520 自転車](#) (2 本の点素パスが取れるかを問う問題です)
- [ARC 039 D 旅行会社高橋君](#) (フローで解く問題ではないですが、「**二重辺連結成分分解**」して得られる各成分は edge-connectivity が 2 以上である (どの 2 点間も 2 本の辺素パスがとれる)」という性質を思うと解法が自然に思えます)

【コメント】

グラフ上の 2 点 s, t が与えられて「頂点を共有しないような $s-t$ パスの集合」を点素パスと呼び、最適な点素パスの集合を求めるタイプの問題です (辺を共有しないものは辺素パス)。こういった問題はほとんどの場合、フローを流す問題になります。

例題 3-5-10 [Evacuation Plan \(POJ No.2175\)](#)

【キーワード】

- フロー
- 最小費用流
- **重みつき**二部マッチング
- 割り当て問題

【AtCoder 上の類題】

- [Maximum-Cup 2013 F 3人の騎士と1匹の犬](#) (2 カテゴリ間の重みつきマッチングを考える点ではよく似ています)
- [Code festival 2014 上海 H Dungeon](#) (これも似た感じですが)
- [ABC 004 D マーブル](#) (想定解法ではないですが最小費用流でも解けます)
- [AOJ 2581 完全順列](#) (面白いですが)
- [AOJ 2293 Dangerous Tower](#) (**任意流量**の最小費用流になります)

【コメント】

例題 3-5-6 と同じく 2 つのカテゴリ間でマッチングを作るタイプの問題 (**割当問題**とも呼びますが) ですが、今度は重みのあるバージョンです。

例題 3-5-11 **The Windy's (POJ No.3686)** <募集中！！！！！！>

【キーワード】

- フロー
- 重みつき二部マッチング
- 割り当て問題
- ノードをコピーしていく発想

【AtCoder 上の類題】

- hoge

【コメント】

基本的には例題 3-5-10 と同じような重みつき二部マッチングの見た目ですが、少し工夫を要するタイプの問題です。各工場について 1 番目, 2 番目, 3 番目, ... とコピーさせて、各おもちゃから見て、「自分がその工場にとって i 番目の到着だったときのコスト」を貼っていくことになります。類似の発想をする問題を見つけられなかったので**募集中**です。

例題 3-5-12 **Intervals (POJ No.3680)**

【キーワード】

- フロー
- DP
- (重みつき) 区間スケジューリング問題
- DP を DAG 上の最短路問題とみなす
- それを容量 K に拡張する

【AtCoder 上の類題】

- [AOJ 2266 Cache Strategy](#) (比較的似ています、少し難しいです)

- [KUPC 2016 H 壁壁壁壁壁壁壁](#) (類題と言っているのか微妙かもですが... [Starry Sky Tree](#) を用いる高速化もするなど難しいです)

【コメント】

非常に中身の濃い問題です。蟻本にて、 $K = 1$ の場合を先に考えていますが、これは (重み付きの) 区間スケジューリング問題で簡単な DP で解けます。

aizuzia さんの [DP を DAG 上の最短経路と思う話](#) を思い出すと、 $K = 1$ の場合の DP を DAG 上の最短経路問題だと思えます。ここから頑張って一般の K に拡張することになります。

例題 3-5-13 その他のフロー

【キーワード】

- フロー

【AtCoder 上の類題】

- [KUPC 2014 I Rain](#) (需要と供給について考えるタイプの最小費用流です)
- [SRM 694 Div1 Hard SRMDiv0Easy](#) (最小流量制約つき最大流です)
- [Japan Alumni Group Summer Camp 2013 Day 4 I Multi Path Story](#) (最小流量制約つき最小費用流です)
- [UTPC 2012 L じょうしょうツリー](#) (想定解法ではないですが**最小費用流問題の双対問題**というタイプの問題になるようです)
- [AOJ 2230 How to Create a Good Game](#) (同じく最小費用流の双対問題です)
- [AOJ 2736 Longest Shortest Path](#) (同じくですが、[Primal-Dual](#) の考え方に根差した解法もあるようです)
- [Japan Alumni Group Spring Contest 2015 E Cost Performance Flow](#) ([Primal-Dual](#) の考え方に根差した解法があります)

【コメント】

蟻本の例題のどれともあまり似てなさそうなタイプのフローを挙げてみました。なお、「3-7 GCJ の問題に挑戦してみよう」にも **DAG の最小パス被覆**な問題があるので、そのタイプの問題はそこで挙げています)。

例題 3-5-14 線形計画問題

【キーワード】

- 線形計画問題
- 単体法

【AtCoder 上の類題】

- [SRM 231 DIV1 Hard Mixture](#) (単体法の verify その 1)
- [UVa 10498 Happiness](#) (単体法の verify その 2)
- [GCJ 2008 Round2 C Simplex](#)

【コメント】

P.223~224 のコラムに関連する話です。

ネットワークフロー問題はだいたい **IP** (整数計画問題) に定式化できます。

それを **LP 緩和** (整数制約のついた変数の整数制約を取り除くこと) します。そうすると最適値は変わってしまうことが多いですが、ネットワークフロー系の問題の多くは**完全単網性**を満たすので、LP 緩和しても最適値が元の問題と変わらないことが言えます。従って、元の問題を LP 緩和した問題に対して単体法なりで解くことにより、元の問題の最適解が求められることが言えます。

少しわかりづらいかも知ですが、この辺の話は以下がとても参考になります:

- [LPとグラフと定式化](#) (とこはるさん)
- [LPっぽいのと最小カット](#) (komiyam さん)
- [競技プログラミングにおける線形計画問題・整数計画問題まとめ](#) (はまやんさん)

類題としては、フローかどうかにかかわらず**単体法**が使える問題を挙げました。

はまやんさんの問題集:

- [競技プログラミングにおける最大流問題まとめ](#)

- 競技プログラミングにおける最小費用流問題まとめ
- 競技プログラミングにおけるマッチング問題まとめ

3-6 平面・空間を扱う "計算幾何"

例題 3-6-1 Jack Straws (POJ No.1127)

【キーワード】

- 計算幾何

【AtCoder 上の類題】

- ABC 002 C 直訴 (三角形の面積です)
- ARC 042 B アリの高橋くん (点と線分の距離です)
- ABC 016 D 一刀両断 (線分と線分の交差判定です)
- AOJ 2661 東大寺 (線分と線分の距離です)
- AOJ 1039 Frisbee Dogs (円や直線の交点です)
- AOJ 2276 ボール (円の接線を求めるなどします)
- AOJ 2316 人魚の魔女 (回す系の問題です)

【コメント】

計算幾何の基本要素を整備していく問題たちです。

例題 3-6-2 White Bird (AOJ 2308)

【キーワード】

- 計算幾何
- ギリギリを考える

【AtCoder 上の類題】

- [AOJ 1133 Circle and Points](#) (ギリギリを考えることで候補を有限個に絞る系です)
- [AOJ 2201 Immortal Jewels](#) (ギリギリを考えます)
- [AOJ 2327 Sky Jump](#) (最適性条件の考察が難しめです)

【コメント】

無限の選択肢のある最適化問題に対して、ギリギリの場合だけ考えればよいという考察をするタイプの問題です。

例題 3-6-3 [Coneology \(POJ No.2932\)](#) <募集中!!!!>

【キーワード】

- 平面走査

【AtCoder 上の類題】

- [Japan Alumni Group Summer Camp 2013 Day 4 D Removing Magical Tiles](#) (結局最大鎖を求める問題になって平面走査します)
- [KUPC 2012 G 村](#) (色んな解法があります)
- [RUPC 2018 day K Move on Ice](#) (典型詰め合わせですがひたすら重いです、平面走査もします)

【コメント】

走査線を動かしていくタイプの問題です。もっと色々あると思うので**募集中**です。

例題 3-6-4 [Beauty Contest \(POJ No.2187\)](#)

【キーワード】

- 凸包

【AtCoder 上の類題】

- [ABC 022 D Big Bang](#) (色んな解法が考えられますが凸包を考えるのは有力です)
- [AGC 021 B Holes](#) (ついこの間の問題です)
- [AOJ 2159 Symmetry](#)
- [AOJ 2442 ConvexCut](#)
- [幾何コンテスト 2013 C 泥棒](#) (動的凸法です)

【コメント】

凸包を考える問題は時として鮮やかですね。もう少し色んな問題がありそうです。

例題 3-6-5 [Intersection of Two Prisms \(AOJ 1313\)](#) <募集中!!!>

【キーワード】

- 数値積分
- シンプソン公式

【AtCoder 上の類題】

- hoge

【コメント】

方針によって明暗が分かると解説がされている問題です。AtCoder 上で見つけれられていないので**募集中**です。

幾何全般の参考記事です:

- [AOJ-ICPC 幾何問題集 \(難易度別\)](#) (うさぎ小屋さん): 非常によい問題集です
- [競技プログラミングにおける幾何問題まとめ](#) (はまやんさん)

3-7 GCJ の問題に挑戦してみよう (2)

GCJ の問題のうち、例題 3-7-3 は **DAG の最小パス被覆** の話題を含むので、それは載せておこうと思います。

例題 3-7-3 Stock Charts (2009 Round2 C)

【キーワード】

- フロー
- DAG の最小パス被覆

【AtCoder 上の類題】

- [AOJ 2279 山](#)
- [AOJ 2251 Merry Christmas](#)
- [AOJ 2352 Divisor](#) (Dilworth の定理を使います)

【コメント】

時々みかける「DAG の最小パス被覆」です。二部グラフの最大マッチングに帰着できます。また、**推移閉包性**を満たした DAG においては

DAG の最大安定集合 = DAG の最小パス被覆

が成立するという **Dilworth の定理**の押さえておきたいところです。

-1 おわりに

蟻本の例題たちを AtCoder の問題に結び付け、さらに類題を加える試みをしました。改良案をドンドン募集しています！



けんちゃん (Otsuki) @drken 

NTTデータ数理システムでリサーチャーをしている大槻です。機械学習やアルゴリズムに関して面白いと思ったことを記事にしていきたいと思います。記事へのリンク等についてはお気軽にしてください大丈夫です。よろしくお願いします。

<http://www.msi.co.jp/>

フォロー



株式会社NTTデータ数理システム

数理科学とコンピュータサイエンスの融合！！

<http://www.msi.co.jp/>

ユーザー登録して、Qiitaをもっと便利に使ってみませんか。

登録する

ログインする





💬 コメント

この記事にコメントはありません。

あなたもコメントしてみませんか :)

ユーザ登録

すでにアカウントを持っている方は[ログイン](#)

How developers code is here.



Qiita

[About](#) [利用規約](#) [プライバシー](#) [ガイドライン](#) [API](#) [ご意見](#) [ヘルプ](#) [広告掲載](#)

Increments

[About](#) [採用情報](#) [ブログ](#) [Qiita Team](#) [Qiita Jobs](#) [Qiita Zine](#)

© 2011-2020 Increments Inc.