@drken 2019年10月27日に更新

AtCoder 版！蟻本 (初級編)

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

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。今回は初級編を扱い、中級編、上級編は別記事に続きます。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、つたじぇー☆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 ～　(本記事が難しいと感じる方へ)
- AtCoder 版！蟻本 (中級編)
- AtCoder 版！蟻本 (上級編)
- AtCoder 版！蟻本 (発展的トピック編)

1 いざチャレンジ！でもその前に --- 準備編

初級編は全体的に、ABC C 問題 (300 点) 相当の難易度の問題が多いです。一番最初に挙げたダーツはかなり難しい (400 点相当) ので、本記事の最後に挑むのも良さそうです。

1-1 プログラミングコンテストって何？

例題 1-1-1　(ハードルの上がった) くじびき　<難問！！！！！>

【キーワード】

- 全探索 (4 重の for 文) だと間に合わない
- 半分全列挙
- 二分探索

【AtCoder 上の類題】

- JOI 2007 本選 C ダーツ　(AOJ 0529 もダーツと同じ問題です。)

【コメント】

蟻本 1-6 に再掲されているハードルの上がったバージョンです。いきなり解くのは難しいかもしれません。ABC や ARC で出題するなら **400 点相当**だと思います。まずは 2-1 章や 2-2 章の比較的易しめの問題を練習して再び挑むのがよさそうです。AtCoder を始めた方が最初に目指す登竜門としてとてもよい問題と言えるでしょう。この問題が自然に解けるようになった頃には大分強くなった実感が湧くと思います！

1-6 気楽にウォーミングアップ

例題 1-6-1　三角形

【キーワード】

- 全探索 (三重の for 文)
- 三角形の成立条件

【AtCoder 上の類題】

- ARC 004 A 2点間距離の最大値　(全探索部分の考え方は一緒)
- ABC 051 B Sum of Three Integers　(よい練習問題です)
- ABC 085 C Otoshidama　(よい練習問題です)

【コメント】

競技プログラミングにおけるすべての基本 **全探索** です。全探索といっても

- for 文を二重三重にして回す
- bit 全探索
- DFS
- BFS

とステップがありますが、これは最も基本的な for 文型の全探索です。ここを抑えるだけでも、ABC の C 問題の打率 3 割までは達成できると思います。

例題 1-6-2　Ants (POJ No.1852)

【キーワード】

- 発想力
- 弾性衝突

【AtCoder 上の類題】

- AGC 013 C Ants on a Circle　(蟻のオマージュにして類似の発想をする問題です...が、ものすごく難しいのでここから下の問題たちをやって実力をつけてから挑むのがよさそうです、はむこさん提供)

【コメント】
言わずと知れた蟻本冒頭の感動的問題です！
一説によると、これが蟻本の名前の由来だという説もあります。

2 基礎からスタート！ --- 初級編

2-1 すべての基本 "全探索"

例題 2-1-1　部分和問題

【キーワード】

- 再帰関数を用いた全探索
- 2^n 通りを調べる全探索 (bit 全探索)

【AtCoder 上の類題】

- ABC 045 C - たくさんの数式　(同じく 2^n 通りを調べる全探索です)
- ABC 079 C - Train Ticket
- ABC 104 C - All Green
- ARC 029 A - 高橋君とお肉
- ABC 002 D - 派閥　(全探索部分の考え方は共通です)

【コメント】

通称 **bit 全探索**と呼ばれているタイプの全探索です。蟻本のように再帰関数の形で書くこともできますし、ビットで集合を管理する手法で全探索することもできます。

- 再帰関数を学ぶと、どんな世界が広がるか
- ビット演算 (bit 演算) の使い方を総特集！ ～ マスクビットから bit DP まで ～

例題 2-1-2　Lake Counting (POJ No.2386)

【キーワード】

- DFS
- (弱) 連結成分分解
- グリッドグラフ上の探索

【AtCoder 上の類題】

- ATC 001 A 深さ優先探索　(かなり似ています)
- ARC 031 B 埋め立て　(比較的似ています)

- ARC 037 B バウムテスト　(見た目は違いますが実装するアルゴリズムは似ています)
- AOJ 1160 島はいくつある？　(ほぼまったくそのままな問題です)

【コメント】

初めて Lake Counting を見たときはとても難しそうに思えましたが、今となってはパターン化した DFS の実装で簡単に記述できます。このタイプの DFS をスラスラと書けるようになることが、競技プログラミングの重要なステップになると思います。

また、ARC 037 B バウムテストは Lake Counting とは見た目は違いますがやることはすごく似ています。こういうのを同じ問題だと思えるようになることもまた、重要なステップかと思います。

- DFS (深さ優先探索) 超入門！ ～ グラフ・アルゴリズムの世界への入口 ～【前編】
- DFS (深さ優先探索) 超入門！ ～ グラフ・アルゴリズムの世界への入口 ～【後編】

例題 2-1-3　迷路の最短路

【キーワード】

- BFS
- BFS を用いる最短路問題
- グリッド上の探索

【AtCoder 上の類題】

- ABC 007 C 幅優先探索　(まんまです)
- JOI 2010 予選 E チーズ　(AOJ 0558 も全く一緒です)
- ABC 088 D Grid Repainting　(予め盤面をいじるタイプの問題は今後よく出て来ます、頭を柔軟にして解いてみましょう)
- AGC 033 A - Darker and Darker　(多点スタートな問題です)
- ARC 005 C 器物損壊！高橋君　(少し発展して、**0-1 BFS** と呼ばれているものです)
- AOJ 0503 Cup　(BFS はパズル的な問題の**最短手数**を求めるのにも有効です)

【コメント】

こちらも超典型テクなので身に着けることがとても重要そうです。

- BFS (幅優先探索) 超入門！ ～ キューを鮮やかに使いこなす ～

例題 2-1-4　特殊な状態の列挙

【キーワード】

- n! 通りの全探索
- next_permutation

【AtCoder 上の類題】

- ABC 054 C One-stroke Path
- JOI 2009 予選 D カード並べ　(AOJ 0546 に同じ)
- yukicoder No.133 カードゲーム　(yumechi さん提供)

【コメント】

小さな n でしか適用できないので難しい問題になると逆に見かけないですが、n! 通りの全探索ができることも重要なステップになります。n が少し大きくなると **bit DP** が想定解法であるケースが多いです。

2-2 猪突猛進！ "貪欲法"

例題 2-2-1　硬貨の問題

【キーワード】

- Greedy
- コイン

【AtCoder 上の類題】

- JOI 2007 予選 A おつり　(ほぼまんまな問題です, AOJ 0521 に同じ)
- AOJ Course コイン問題　(より一般的なコイン問題、おそらく Greedy は通らないです)

【コメント】

硬貨の問題が Greedy で良いことは、そんなに自明じゃないようです。

例題 2-2-2　区間スケジューリング問題

【キーワード】

- Greedy
- 区間の終端でソート (DEGwer さんの数え上げテクニック集 にも記載あり)

【AtCoder 上の類題】

- KUPC 2015 A 東京都　(見た目文字列な問題ですが、区間スケジューリングです)
- ABC 103 D - Islands War　(実は区間スケジューリングです)
- Codeforces 296 DIV1 B Clique Problem　(少し難しめ、区間スケジューリング問題に帰着できます)
- ABC 038 D プレゼント　(難しめです、区間ソートして LIS に帰着)

【コメント】

区間の終端 (または始端) でソートするのは極めてよくみるテクニックで、今後難しい問題に挑むときにも常に念頭に置いておきたいです。

例題 2-2-3　Best Cow Line (POJ No.3617)

【キーワード】

- Greedy
- 辞書順最小なものを求める Greedy

【AtCoder 上の類題】

- ABC 076 C Dubious Document 2　(辞書順最小 Greedy の練習です)
- ABC 007 B 辞書式順序　(そもそも辞書次式順序とはなにか)
- ABC 009 C 辞書式順序ふたたび　(かなり難しいです)

【コメント】

辞書順最小なものを求めよと言われたときにとにかく先頭から順に最小になることを優先していく考え方は超頻出ですね。

例題 2-2-4　Saruman's Army (POJ No.3069)

【キーワード】

- Greedy
- より厳しいところをとっていく Greedy (より一般に「**交換しても悪化しない**」)

【AtCoder 上の類題】

- ABC 083 C Multiple Gift　(例題はなるべく遠い方を選ぶ Greedy でしたが、これは近い方を選ぶ Greedy です)
- ARC 006 C 積み重ね　(典型問題です)

- ABC 005 C おいしいたこ焼きの売り方　(少し難しくなった Greedy です、Greedy に解けるマッチングの例です)
- SRM 560 DIV1 Easy TomekPhone　(一次元量同士を比較する最大二部マッチング、Greedy に解けるマッチング例です)
- ABC 091 C 2D Plane 2N Points　(二次元量同士を比較する最大二部マッチング、Greedy に解けるマッチング例です)

【コメント】

Greedy アルゴリズムを考えるときに、より厳しいところをとっていく考え方は頻出のイメージです。その最も典型的な例として、一次元的 (二次元量もOK) な数量の大小関係だけでマッチング条件が決まるような問題における最大二部マッチング問題があります。

例題 2-2-5　Fence Repair (POJ No.3253)

【キーワード】

- Greedy
- priority_queue
- ハフマン符号
- 二分木

【AtCoder 上の類題】

- Codeforces 263 DIV2 C Appleman and Toastman　(ヘクトさん提供)

【コメント】

ハフマン符号を求める Greedy、priority_queue も用います。

Codeforces の問題ですが、なんとか近いものがありました！

2-3 値を覚えて再利用 "動的計画法"

例題 2-3-1　01ナップサック問題

【キーワード】

- DP
- ナップサック DP

【AtCoder 上の類題】

- AOJ Course 0-1ナップザック問題　(例題そのものです)
- TDPC A コンテスト　(部分和問題です)
- TDPC D サイコロ　(添字増やす系のナップサックです)
- ABC 015 D 高橋くんの苦悩　(全体の個数制約が加わります、くちもちとくらさん提供)
- JOI 2012 予選 D 暑い日々　(AOJ 0579 に同じ)
- TDPC E 数　(いわゆる**桁 DP** です。桁 DP もナップサックの一種で、bit DP とは別物です)

【コメント】

ナップサックな DP については記事を書いてみました。DP を漸化式だととらえる立場からの入門記事です。DP を全探索の効率化から入る立場の入門資料として「プログラミングコンテストにおける動的計画法」がとてもよいです。漸化式派と全探索メモ化派は、強い人たちの間でも二分されているので肌の合う考え方で入門していくのがいいと思います。

桁 DP について参考記事:

- Digit DP 入門 (torus711 さん)
- 競技プログラミングにおける桁DP問題まとめ (はまやんさん)

例題 2-3-2　最長共通部分列問題

【キーワード】

- DP
- LCS
- 二次元ナップサック DP

【AtCoder 上の類題】

- AOJ Course Longest Common Subsequence　(例題そのものです、nadare さん提供)
- AOJ Course Edit Distance (Levenshtein Distance)　(編集距離)
- Indeedなう C Optimal Recommendations　(今度は三次元です)

【コメント】

系列に沿って進んでいく index が 2 つになったバージョンのナップサック型 DP です。

例題 2-3-3　個数制限なしナップサック問題

【キーワード】

- DP
- ナップサック DP
- 個数制限なしナップサック DP

【AtCoder 上の類題】

- AOJ Course ナップザック問題　(例題そのものです)
- AOJ 2502 VOCAL ANDROID

【コメント】

各品物を何個でも選んでよいバージョンのナップサックです。愚直にやると計算量が $O(nW^2)$ かかってしまうところをうまくやって $O(nW)$ に落とすテクニックです。**配列再利用**と呼ばれる DP 時のメモリ消費を抑えるテクニックとも相性がいいです。忘れた頃に見かけるイメージです。

例題 2-3-4　01ナップサック問題その２

【キーワード】

- DP
- ナップサック DP
- dp[W] := 重み W 以下での価値の最大値 -> dp[V] := 価値 V 以上を達成できる重さの最小値

【AtCoder 上の類題】

- ABC 032 D ナップサック問題　(後に出て来る**半分全列挙**も含みます)
- ARC 057 B 高橋君ゲーム
- AGC 033 D - Complexity　(1000 点問題でとても難しいですが、言われれば「あっ！」となる非自明な一発ネタ問題でした)

【コメント】

このテクニックも忘れた頃に見かけるイメージがあります。

例題 2-3-5　個数制限付き部分和問題

【キーワード】

- DP
- ナップサック DP
- DP の値に bool 値より多くの情報を持たせる
- DP の値に復元のための情報を持たせる
- (少し発展すると) 戻す DP

【AtCoder 上の類題】

- Maximum-Cup 2018 D Many Go Round　(DP に bool 値より多くの情報を持たせることで解けます、bitset 高速化で殴ることもできます)
- JOI 2013 本選 D フクロモモンガ　(少し難しめです、DP 値を利用して状態を復元します、AOJ 0601 に同じ)
- ARC 083 E Bichrome Tree　(難しめです)
- ARC 028 D 注文の多い高橋商店　(かなり難しいです、戻す DP とのセット)

【コメント】

DP の値を単に bool 型にするのではなく、より多くの情報を持たせることで問題を解いたり、さらに DP 値も利用して状態を復元したりする系の問題です。その辺りのことは前記事に詳しく書いてみました。元々高度なテクニックなので問題例も難しめです。

戻す DP について参考記事:

- 戻すDP (sigma さん)
- 競技プログラミングにおける戻すDP問題まとめ (はまやんさん)

例題 2-3-6　最長増加部分列問題

【キーワード】

- DP
- LIS
- インライン DP (実家 DP)

【AtCoder 上の類題】

- ABC 006 D トランプ挿入ソート
- ABC 038 D プレゼント　(区間スケジューリング問題のところでも挙げました)
- TDPC K ターゲット
- JOI 2016 春合宿 day1-1 マトリョーシカ人形　(難しいです。LIS ではないですが LIS の理解があると解きやすいです。問題文はここ)

【コメント】

sky さんの提唱するインライン DP (通称、実家DP) の最も簡単な例で、なおかつ超頻出の **LIS** です。LIS に帰着できる問題は数多いのでライブラリとして持っておくだけでも強いですが、LIS の仕組みにより精通しているとより応用の効く問題もあります。

参考記事:
競技プログラミングにおける最長部分増加列問題まとめ (はまやんさん)

例題 2-3-7　分割数

【キーワード】

- DP
- 分割数

【AtCoder 上の類題】

- yukicoder No.269 見栄っ張りの募金活動
- 第4回 ドワンゴからの挑戦状 予選 C - Kill/Death
- JOI 2008 予選 F ビンゴ (AOJ 0537 に同じ)

【コメント】

時々高難易度な問題で部分的に登場する分割数です。同じように高難易度な問題で登場する話題に、スターリング数やベル数があります。

参考記事:

- 分割数と、分割数が使える問題まとめ
- 競技プログラミングにおける数学的問題まとめの写像12相部分 (はまやんさん)

例題 2-3-8　重複組合せ

【キーワード】

- DP
- 重複組み合わせ

【AtCoder 上の類題】

- ABC 110 D - Factorization　(とても教育的な良問でした)
- ABC 021 D - 多重ループ　(重複組合せぴったりです)
- ARC 004 D - 表現の自由　(現代ではすっかり練習問題レベルになりましたね)

【コメント】

難しい問題の部分的なパートで登場するイメージのある重複組合せです。確かこの蟻本の例題をさらに応用した問題があった気がするのですが、誰かご存知でしょうか...？？？

2-3-α その他典型としてあると思う動的計画法 (bit DP, Trie DP, Grundy DP, 行列累乗, 実家 DP, 連立方程式, Convex Hull Trick は中級編や上級編にあるので除きます)

だいたい、TDPC です。

例題 2-3-9　ゲーム DP

【キーワード】

- DP
- ゲーム DP

【AtCoder 上の類題】

- TDPC B ゲーム
- ABC 025 C 双子と○×ゲーム
- ABC 078 D ABS

【コメント】
ゲーム DP は蟻本上級編に載っているのですが、ゲーム DP (ゲーム木探索) だけなら初級編の段階で学んでもよさそうです。(TDPC は B で詰まるという声は多々聞くのでよい解説を見つけたいです)

例題 2-3-10　区間 DP

【キーワード】

- DP
- 区間 DP

【AtCoder 上の類題】

- TDPC I イウイ
- AOJ 2415 刺身　(Monge です)
- ATC 002 C 最適二分探索木　(刺身と一緒ですが Monge してもなお足りず、Hu-Tucker が必要です)

【コメント】

区間に対する DP です。Monge 性は蟻本にない話題なのでどこかで学ぶ必要があります。通常の区間 DP は O(n^3) かかり、Monge 性を満たすと O(n^2) にすることができます。

最適二分探索木問題に限ってはさらに O(n logn) で解ける Hu-Tucker のアルゴリズムが知られています。

参考記事:
競技プログラミングにおける区間DP問題まとめ (はまやんさん)

例題 2-3-11　ツリー DP

【キーワード】

- DP
- ツリー DP

【AtCoder 上の類題】

- ABC 036 D 塗り絵
- ABC 070 D Transit Tree Path
- TDPC N 木　(少し難しめです)
- TDPC P うなぎ　(**ツリー二重 DP** と呼んでいます。ノード v の値の更新自体に DP が必要なやつです。)
- ARC 083 E Bichrome Tree　(ツリー二重 DP の他例です。例題 2-3-5 の類題にも挙げました。)

【コメント】

ツリー上の DP です。より高度な話題として、全方位木 DP があります。

参考記事:

競技プログラミングにおける木DP問題まとめ (はまやんさん)

競技プログラミングにおける全方位木DP問題まとめ (はまやんさん)

例題 2-3-12　DAG DP

【キーワード】

- DP
- DAG DP

【AtCoder 上の類題】

- ABC 037 D 経路
- JOI 2006 本選 D 最悪の記者　(AOJ 0519 に同じ)

【コメント】

ツリー DP を発展させて、一般的な DAG 上での DP です。ツリー DP とほぼ変わらない実装でできます。

例題 2-3-13　グリッド DP

【キーワード】

- DP
- グリッド DP

【AtCoder 上の類題】

- JOI 2006 予選 F 通学経路　(AOJ 0515 に同じ)
- JOI 2007 本選 D ぴょんぴょん川渡り　(AOJ 0530 に同じ)
- JOI 2008 本選 D 散歩　(AOJ 0541 に同じ)
- JOI 2009 予選 E 通勤経路　(AOJ 0547 に同じ)

【コメント】

DAG DP の一種ですが、特にグリッド上の DP です。JOI ではよく見るイメージです。
一般の DAG DP と異なり、メモ化再帰じゃなくても for 文でループを回す実装ができます。

例題 2-3-14　ソートしてからナップサック DP

【キーワード】

- DP
- ナップサック DP
- 先にソート

【AtCoder 上の類題】

- JOI 2011 予選 C 最高のピザ　(AOJ 0567 に同じ)
- JOI 2010 本選 B 古本屋　(AOJ 0561 に同じ)
- TDPC H ナップザック
- 2017 CODE FESTIVAL Final D Zabuton　(難しいですが、先にソートするんだろうな... と思えることが重要です。そう思えたらどうソートするかをひたすら考えます)
- SRM 502 DIV1 Medium TheProgrammingContestDivOne　(SRM の問題ですが、個人的に良問だと思います。)

【コメント】

DP 部分はごく普通のナップサック DP ですが、一見すると順序も自由に入れ替えられるので困るタイプの DP です。こういったものは順番を先に fix できるケースが多いのでそれをどうしたらいいかをひたすら考察することになります。

DEGwer さんの数え上げテクニック集の「3.1 大きい順に並べる」にも類似の記載があります。

余談ですが「ソートしてナップサック」をかなり一般的なフレームワークに乗せた論文がアルゴリズム系の国際会議 ISSAC 2016 の Best Paper Award に選ばれています！

- Yasushi Kawase, Kazuhisa Makino, and Kento Seimi: Optimal Composition Ordering Problems for Piecewise Linear Functions, Proceedings of the 27th International Symposium on Algorithms and Computation (ISAAC2016, LIPICS64)

例題 2-3-15　部分文字列 DP

【キーワード】

- DP
- 部分文字列 DP

【AtCoder 上の類題】

- ARC 081 E Don't Be a Subsequence　(難しいですが典型的な部分文字列 DP です)
- AOJ 2895 回文部分列 (AUPC 2018 day3 G)　(CSA で既出だったようですが面白い問題でした)
- AOJ 1392 Shortest Common Non-Subsequence (ICPC Asia 2018 D)　(ついこの間の ICPC アジアの問題です)
- TDPC G 辞書順　(典型的ではありますが、復元パートがかなり思いつきづらく難しいです)

【コメント】

特に名前がないので勝手に名付けました！
文字列の部分文字列全体を探索するような DP です。桁 DP と同様ちゃんと名前がつければ、もう少し解法が広まる気がします。

参考記事:

部分列 DP --- 文字列の部分文字列を重複なく走査する DP の特集

例題 2-3-16　挿入 DP

【キーワード】

- DP
- 挿入 DP

【AtCoder 上の類題】

- TDPC O 文字列
- 天下一プログラマーコンテスト 2015 本戦 E 天下一コップ
- JOI 2018/2019 予選 F - 座席 (Seats)
- JOI 2016 春合宿 day1-3 ソリティア　(かなり難しいです、問題はここ)

【コメント】

発想はナップサックに近いのですが、今ある並びの中に新しいものを挿入していく DP です。高難易度でお馴染みです。「bit DP でやりたくなるけど制約上とても無理で、部分点として bit DP が設定されている」というのがよくあるパターンです。

DEGwer さんの数え上げテクニック集の「3.2 順列は挿入 DP」に挿入 DP がどういうときに有効かの説明が書いてあります。

参考記事:

競技プログラミングにおける挿入DP問題まとめ (はまやんさん)

例題 2-3-17　連結性 DP

【キーワード】

- DP
- 連結性 DP

【AtCoder 上の類題】

- TDPC S マス目

【コメント】

超面倒な DP です。

参考記事:

競技プログラミングにおける連結DP問題まとめ (はまやんさん)

例題 2-3-18　きたまさ法

【キーワード】

- DP
- きたまさ法

【AtCoder 上の類題】

- TDPC T フィボナッチ

【コメント】

きたまさ法です。蟻本 P.182 のコラム「もっと高速な漸化式の計算」に「興味のある人は考えてみるといいでしょう」と書いてあるものです。

DP のその他参考資料

- 競技プログラミングにおける動的計画法問題まとめ (はまやんさん)
  勉強の指針になります
- 数え上げテクニック集 (DEGwerさ ん)
  DP に限らないです。すごいです
- DP (はむこさん)
  非常にマニアックな DP テクニックへのポインタが充実しています

2-4 データを工夫して記憶する "データ構造"

例題 2-4-1　Expedition (POJ No.2431)

【キーワード】

- priority_queue
- 「あとで補償する」という発想

【AtCoder 上の類題】

- 2017 CODE FESTIVAL THANKS C - Factory　(priority_queue を有効活用できる例です)
- ABC 062 D - 3N Numbers　(priority_queue)
- JOI 2009 本選 E ダンジョン　(超高難易度問題ですが「あとで補償する」という発想は似ています, AOJ 0553 に同じ)

【コメント】
「あとで補償する」という発想は高難易度な問題でよく見るイメージです。そして大体そういう問題はセグメントツリーや BIT を使って効率的に実行していくイメージがあります。蟻本の例題 Expedition はそういう問題にしては珍しく高度なデータ構造を必要としない問題だと言えます。

例題 2-4-2　二分探索木 (set, map の練習)

【キーワード】

- set, map

【AtCoder 上の類題】

- ABC 085 B Kagami Mochi　(set を使えると簡単です)
- ABC 091 B Two Colors Card Game　(map を使えると簡単です)

【コメント】

ABC C 問題を解けるようにしていく上で set や map を使えるようになることも重要なステップだと思います。

例題 2-4-3　食物連鎖 (POJ No.1182)

【キーワード】

- Union-Find 木
- ノードコピーして考える

【AtCoder 上の類題】

- ATC 001 B Union Find　(とりあえず Union-Find 木を verify する問題です)
- ABC 049 D 連結　(Union-Find 木のよい例題です)
- ARC 097 D Equals　(やはりよい例題です)
- ARC 036 D 偶数メートル　(難しめですが、蟻本の例題に非常に近い問題です)

- ABC 087 D People on a Line　(色んな解法がありますが、**重み付き Union-Find 木**が楽です)
- 2017 CODE FESTIVAL THANKS H Union Sets　(非常に色んな解法のある問題です、**永続 Union-Find 木**や**並列二分探索**で通せます)
- JOI 2011 本選 E JOI 国のお祭り事情 (Festivals in JOI Kingdom)　(大分難しいです, AOJ 0575 に同じ)

【コメント】

Union-Find 木はちょくちょく出てくるデータ構造です。蟻本の例題はとてもよい問題ではあるのですが、中国語の問題というのが大分辛いですね...。発展的話題として、重み付き Union-Find 木や、永続 Union-Find 木があります。

参考記事:
競技プログラミングにおけるUnionFind問題まとめ (はまやんさん)

2-5 あれもこれも実は "グラフ"

例題 2-5-1　二部グラフ判定

【キーワード】

- DFS
- 二部グラフ
- 二部グラフ判定 (2色彩色)

【AtCoder 上の類題】

- AtCoder ABC 126 D - Even Relation　(二部グラフになるように白黒の塗り分けを行う問題です)
- CODE FESTIVAL 2017 qualB C 3 Steps　(よさげな問題です)

- Maximum-Cup 2018 C 嘘つきな天使たち　(二部グラフ判定で殴れますが、Union-Find 木を用いるともっと楽です)
- ARC 041 D 辺彩色　(難しめです)
- AOJ 2370 RabbitWalking　(かなり難しめです)

【コメント】

いずれも「二部グラフ ⇔ 奇数長サイクルを含まない」を活用する問題ですね。

例題 2-5-2　Roadblocks (POJ No.3255)

【キーワード】

- 最短路問題
- Dijkstra のアルゴリズム

【AtCoder 上の類題】

- SoundHound 2018 予選 D - Saving Snuuk　(海外対応以降の AtCoder では数少ない純粋な Dijkstra 法の問題です)
- JOI 2007 予選 F 船旅　(グラフの形が動的に変わりますが普通の Dijkstra と同様です, AOJ 0526 に同じ)
- ABC 035 D トレジャーハント
- ARC 005 C 器物損壊！高橋君　(**0-1 BFS** ですが、Dijkstra でもできます)
- WUPC 2012 会場への道　(**拡張 Dijkstra** とよく言われているタイプです)
- ARC 084 D Small Multiple　(ABC 史上最難問と名高い問題です, BFS でもよいです)

【コメント】

お馴染みの Dijkstra 法です！

意外と純粋な Dijkstra な問題はなかなかないですね... AOJ には多数あるのですが...

例題 2-5-3　Conscription (POJ No.3723)

【キーワード】

- 最小全域木問題
- Kruskal のアルゴリズム

【AtCoder 上の類題】

- AOJ Course 最小全域木　(ライブラリの verify 用に)
- ABC 065 D Built?　(最小全域木に関する理解を問う良問です)
- 2015 IndeedなうB D Game on a Grid
- JAG Practice Contest for ACM-ICPC Asia Regional 2012 C Median Tree　(きっと Kruskal だろうと当たりはつきますが、是非ともそれでいい理由を抑えたいところです)
- AOJ 1350 There is No Alternative　(Kruskal 法が成立する理由をよく理解すれば解けます)
- UTPC 2013 F 魔法の糸　(少し難しめです)
- ARC 093 E Bichrome Spanning Tree　(難しめですが、Kruskal 法の最適性条件についての深い理解を問う良問です)
- ARC 018 D 僕は友達が少ない　(かなり難しいです)
- CODE FESTIVAL 2017 Final (Parallel) J Tree MST　(激ムズですが是非載せておきたいです。実は元のグラフがツリーでなくても解けることを、とこはるさんから聞きました)

【コメント】

最小全域木絡みは良問が多いですね！

調べれば調べるほど気分がよくなって、ついつい色んな問題を載せてしまいました。

参考記事:

様々な最小全域木 (前原さん): 面白いです！

例題 2-5-4　Layout (POJ No.3169)

【キーワード】

- 牛ゲー
- 最短路問題
- Bellman-Ford のアルゴリズム

【AtCoder 上の類題】

- AOJ Course 単一始点最短経路（負の重みをもつ辺を含む）　(Bellman-Ford 法ライブラリの verify 用です)
- ABC 061 D Score Attack　(Bellman-Ford 法が使える問題です)
- UTPC 2013 H Asteroids2　(牛ゲーです)
- 東京工業大学プログラミングコンテスト 2015 N 何かグラフの問題　(当初の想定は**最小平均長閉路**だったようです)

【コメント】

出ました！！！！！いわゆる**牛ゲー**です！！！！

蟻本初級編で最も難しいと呼び声の高い牛ゲーです！！！

ちなみに牛ゲーの名前の由来がまさにこの POJ の例題に牛が登場するからです。

参考記事:

競技プログラミングにおけるベルマンフォード問題まとめ (はまやんさん)

例題 2-5-5　Warshall-Floyd を使う問題

【キーワード】

- 全頂点間最短路
- Warshall-Floyd のアルゴリズム

【AtCoder 上の類題】

- ABC 012 D バスと避けられない運命
- ABC 073 D joisino's travel
- ABC 079 D Wall

【コメント】

蟻本には例題がないですが需要はありそうなので、Warshall-Floyd のアルゴリズムが使える問題を挙げました。

2-6 数学的な問題を解くコツ

例題 2-6-1　線分上の格子点の個数

【キーワード】

- 最大公約数
- Euclid の互除法

【AtCoder 上の類題】

- AOJ 0583 公約数　(最大公約数ライブラリの verify に)
- ABC 070 C Multiple Clocks　(最小公倍数を求めます、ライブラリの verify に使えそうです)
- AGC 001 B Mysterious Light　(比較的発想は似ています)
- CSA 068 DIV2 B Integer Coords　(CSA ですがまんまな問題です)
- AGC 018 A Getting Difference　(最大公約数に関する深い理解が求められます)
- JOI 2006 本選 E 最軽量のモビール　(少し難しめです, AOJ 0520 に同じ)

【コメント】

ユークリッドの互除法自体は頻出ですが、「線分上の格子点の個数」が最大公約数を求

めることで求められるというのは、少し理解が難しいかもしれません。

例題 2-6-2　双六

【キーワード】

- 拡張 Euclid の互除法

【AtCoder 上の類題】

- AOJ Course NTL_1_E - 拡張ユークリッドの互除法　(拡張ユークリッドの互除法そのものを試せる問題です)
- ARC 067 E Grouping　(とりあえず mod_inverse を使いそうな例です)
- SRM 537 DIV1 Easy KingXNewCurrency

【コメント】

AtCoder 上のピッタリな問題は見つけられなかったです。募集中です。
とりあえず拡張 Euclid の互除法を確かめようとなったら、mod_inverse を拡張 Euclid 互除法によって実装してみて、mod_inverse が登場する問題を解くのがよさそうです。

例題 2-6-3　素数判定

【キーワード】

- 素数
- 素数判定
- 素因数分解

【AtCoder 上の類題】

- ARC 017 A 素数、コンテスト、素数　(素数判定の verify に)
- ABC 052 C Factors of Factorial　(素因数分解の verify に)
- ABC 110 D - Factorization　(素因数分解 + 重複組合せ、とても教育的な良問でした)

【コメント】

おなじみの素数判定 & 素因数分解です。

例題 2-6-4　素数の個数

【キーワード】

- 素数
- Eratosthenes の篩

【AtCoder 上の類題】

- 天下一プログラマーコンテスト2012 予選C A 与えられた数より小さい素数の個数について
- ABC 084 D 2017-like Number　(累積和テクも使います)
- AOJ 0009 素数　(素数関連のライブラリの verify にはこの問題が一番よいかもです)
- AOJ 2286 Farey Sequence　(一見違う問題ですが、エラトスネテスの篩っぽい方法で解けます)

【コメント】

そすーそすー

Eratosthenes の篩です。

Eratosthenes の篩的な前処理をしたあとに「素因数分解」を高速に実行する osa_k 法も。

例題 2-6-5　区間内の素数の個数

【キーワード】

- 素数
- Eratosthenes の篩
- Eratosthenes の区間篩

【AtCoder 上の類題】

- JAG Practice Contest for ACM-ICPC Asia Regional 2017 C Prime-Factor Prime

【コメント】

英語ですがなんとか区間篩の問題がありました。日本語の問題も募集中です。

例題 2-6-6　Carmichael Numbers (Uva No.10006)

【キーワード】

- べき
- 繰り返し二乗法
- ダブリング
- Fermat の小定理
- カーマイケル数 (擬素数)

【AtCoder 上の類題】

- JOI 2007 春合宿 fermat フェルマー方程式　(繰り返し二乗法の verify に使えそうです。問題文はここ)
- ARC 020 C A mod B Problem　(かなり難しめです、ダブリングの応用です)

- Japan Alumni Group Summer Camp 2015 Day 4 D Identity Function　(かなり難しめです, 初等整数論の知見を総動員します)

【コメント】

カーマイケル数大好きです！！！！！！

最後の問題の原案を担当していましたが、元ネタはカーマイケル数です。

参考記事:

- 整数テクニック集 (kirika さん)
- 競技プログラミングにおける数学的問題まとめ (はまやんさん)

-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てみました。改良案をドンドン募集しています！

けんちょん (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いただいて大丈夫です。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/

コメント

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