普通の整列ではない並べ替えって？ ―立命館大学の2025年6月公表試作問題II―

情報処理学会・学会誌「情報処理」
2025年8月1日 10:39

久野　靖（電気通信大学）

目次

「教科『情報』の入学試験問題って？」，2025年は「はじめの年」として，共通テストやこの年から情報科の個別入試を実施した大学では「過去問」ができて，2年目がどうなるか関心が高いところです．

一方で，2025年は見送ったけれど2026年個別入試に「情報」を出題する大学として，高知工科大学，大阪工業大学などが挙げられます．その中から今回は，立命館大学が2025年6月18日に公開した，「情報」科目の試作問題の大問IIを取り上げましょう．

実は，この問題がどんな問題かということについて，個人的にはかなり興味があったのです．というのは，同大学のWebページ「一般選抜 『情報』の本学独自入試の導入について」の「試作問題の問題構成と出題方針」で大問IIについて次のように書かれていたからです．

> II 並べ替えの基本概念，およびそれを実現するアルゴリズムを考えさせることで，論理的思考力を問う．

https://ritsnet.ritsumei.jp/admission/general/Informatics.html#a2

え，並べ替えって，データを大小順に並べる，整列（ソート）ですよね．そのアルゴリズムを考えるって，新しい整列アルゴリズムを考案するんですか……と思ったとか思わないとか．

それはさておき，全問題や基本方針などは同大学のサイト[1) 2)]にありますので，必要な方は参照してください．では，大問IIの解説を始めましょう．

「並べ替え」とは？

さっそく，問題文を読んでみます．

以下の図１のように十分長い一直線状の線路があり，途中に本線と副線に分岐している箇所があるとする。この線路上を列車は右端から左端に走る（逆走はできない）。各列車にはアルファベット１文字でそれぞれ異なる列車名が付けられており，列車同士は十分な間隔をあけて走る。以下，本線と副線に分岐した箇所は１列車分の長さしかない。また副線上で列車は停止できるが，本線上では列車は停止せずに通過するものとする。

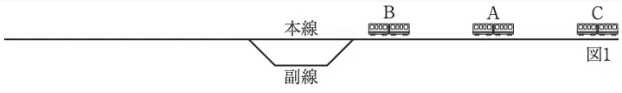

図1

例えば図１のように線路の右端からB，A，Cの並び順で列車が走ってきた場合，先頭にあった列車Bを図２のように副線に退避させてから，後続の列車Aを図３のように本線を通じて通過させ，その後図４のように列車Bを発車させてから，後続の列車Cを図４～図５のように本線を通じて通過させれば，線路の左端では列車の並び順をA，B，Cにすることができる。

これについて以下の各問いに答えよ。

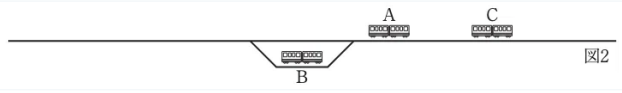

図2

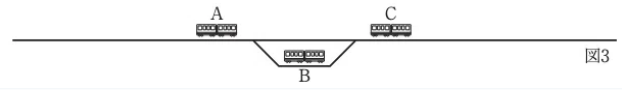

図3

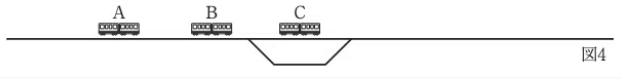

図4

図5

なんと！ 「並べ替え」とは，列車の並び順を入れ替えることのようです．そのつもりで見ると，確かに列車の並べ替えを実現するアルゴリズムをこの先で扱うので，先の基本方針は正しいです．それにしても，立命館大学の中の人に一杯食わされた気分です（が，後で出てくるように，それは私の一人相撲だったようです）．

副線に退避する列車は？

ではさっそく，簡単な問題からやってみましょう．まず，この大問IIの解答指示があります．

次の文章中の空欄 ア ～ ウ に入れるのに適当なものを，後の選択肢のうちからすべて選べ。 エ オ に当てはまる適当な数値または式を，解答用紙の所定の欄に記入せよ。

次に，一番簡単な（と思われる）[1]の問題文と選択肢を示します．

〔1〕 線路の右端での列車の並び順がB，A，D，Cのとき，これらがすべて線路の左端に到達した時の並び順をA，B，C，Dにしたい。このとき4つの列車のうち副線に退避することになる列車をすべて選ぶと ア となる。

ア の選択肢

⓪ A　① B　② C　③ D

大事なのは，本線では止まれないので，追い越させるためには必ず副線で退避する必要があることです．B，A，D，CがA，B，C，Dになるので，Bは必ずAに追い越されなければなりませんし，Dも必ずCに追い越されなければなりません．そして，その2つ以外の追い越しは必要ありません．したがって，解答は①B，③Dということになります．

1つの副線で昇順の列にできるのは？

次の問題は，少し難しくなります．その前に，「A，B，C，D」とか「A，B，C，D，E」とか何回も書くのは大変なので，そのような……Aから始まりアルファベット順にいくつか並んでいる……ものを

「昇順の列」と（この解説限りで）呼びましょう．

そして，［2］の問題文と選択肢は次のとおりです．

〔2〕 線路の右端での列車の並び順が［イ］だったとき，列車がすべて線路の左端に到達した時の並び順をA，B，C，D，Eにできる。［イ］に入れるのに適当なものをすべて選べ。

［イ］の選択肢

⓪ A，D，C，B，E　　① C，A，B，E，D

② C，A，D，B，E　　③ B，A，E，C，D

どうでしょうか．副線は1カ所で，1列車しか入れないことがポイントです．たとえば①は，Cが退避している間に，A，Bが通過し，そこでCが出て，次にEが退避してDが通過，でできますね．③も，Bが退避してAが通過，Bが出て，Eが退避，C，D通過，でOKです．

残りはどうでしょう．⓪は，Dが退避してB，Cを通過させる間に，Cも退避してBを通過させたいので，副線の利用がぶつかってしまいます．②は，Cが退避してA，Bを通過させる間に，Dも退避してCが出たあとに出なければなりませんから，これも副線の利用がぶつかってしまいます．したがって，解答は①，③ということになります．

図を使って定式化してみましょう．昇順になっていない，自分より前の字が後ろ（直後とは限りません）にある文字からから矢線を引き出し，自分より前の文字すべてが終わった直後に線を引きます．

こうして見ると，①，③のように矢線がどの区間でもせいぜい1本ならOK，⓪，②のように矢線が2本以上の区間があればだめ，のようです．2本以上あるとは，追い越し中にまた別の追い越しをする，ということですから，当然ですね．

2つの副線で昇順の列にできるのは？

次の問題は，副線が2つある場合です．問題，図，選択肢を示します．

〔3〕 図6のように一直線状の線路上に副線に分岐する箇所が2か所あり，これらの間の距離は十分離れているものとする。このとき，線路の右端での列車の並び順が［ウ］だったとき，列車がすべて線路の左端に到達した時の並び順をA，B，C，D，Eにできる。［ウ］に入れるのに適当なものをすべて選べ。

図6

［ウ］の選択肢

⓪ E，D，A，B，C　　① A，D，C，B，E

② D，B，A，E，C　　③ E，D，A，C，B

さっきと同様，矢線を引いてみましょう．

⓪ E，D，A，B，C　　① A，D，C，B，E

② D，B，A，E，C　　③ E，D，A，C，B

今度は，副線への分岐が2カ所あるので，矢線が2本までの区間しかない⓪，①，②はOKです．たとえば，一番込み入っている②の場合，1本目の分岐でBをAの後ろ，EをCの後ろに移動させ，2本目の分岐でDをCの後ろに移動させます（または，1本目と2本目の役割を入れ替えてもよい）．

一方，③はどうでしょう．CをBの後ろ，DをCの後ろ，EをDの後ろにみんなしなければならないので，分岐が2カ所では足りません（3カ所ならできます）．したがって，解答は⓪，①，②ということになります．

もっと多くの副線で昇順の列に，何本？

次の問題は，最初の並び順を与えて，分岐が何個所必要かを問うものです．この問題からは，選択肢ではなく，解答欄に答えを記入します．

〔4〕 一直線状の線路上に副線に分岐する箇所がさらに多く存在する場合を考える。線路の右端での列車の並び順がB，C，D，E，A，Fのとき，列車がすべて線路の左端に到達した時の並び順をA，B，C，D，E，Fにできるためには，副線に分岐する箇所は少なくとも［エ］箇所必要である。

例によって，矢線を引いてみます．

Aが終わりから2番目にあるので，前の文字の矢線はそのAの後ろまで伸ばすしかありませんから，こうなるわけです．分岐する個所の数も同じです．したがって，解答は「4」カ所ということになります．

列車の数がnだと，最大何本？

最後の問題は，前の問題を踏まえて一般化します．この問題も，解答欄に答えを記入します．

〔5〕 線路の右端から進入する列車の数が n であるとする（但し，n は26以下とする）。線路の右端での列車の並び順がどのようなものであっても線路の左端に到達した時の並び順を必ずアルファベット順にできるためには，副線に分岐する箇所は少なくとも［オ］箇所必要である。［オ］は n を用いて答えよ。

これまでの（2問目以降の）どの問題もそうですが，この問題も前の問題がヒントになります．前の問題の矢線の図をよく見てください．Aが後ろの方にあると，その前の区間では前にあるすべての文字の矢線が同時に描かれることが分かります．

ですから，「どのようなものであっても」，つまり本数が最大の場合を知りたければ，Aが「最後に」ある場合を考えればよいわけです．そのとき，Aより前にあるすべての矢線が同時に描かれることになります．その本数は，Aを除くすべての文字に対応します．分岐する個所の数も同じです．したがって，解答は「n - 1」個所ということになります．

一応，n = 6 でAが最後の図を2通り描いてみました，Aが最後なら，A以外の全部の文字からの矢線が最後まで引かれるので，n - 1 = 5 本の矢線で，副線も5本必要です．

n - 1 であって n - 2（以下）ではないことの説明も一応しましょうか．図を見れば明らかなように，n がいくつでも，矢線が n - 1 本引けるので（Aを除く各文字ごとに引ける），n - 2（以下）の副線では足りないと分かります．

おまけ1:「矢線引き」をプログラムに

こうして見ると、この問題の大部分は「最初の並びに対して矢線を引く」ことで解けるわけです．それを先には紙とペンでやりましたが，もう1段進めて，プログラムにやらせるのはどうでしょうか．

そんな大変な，と思うかもしれませんが，10行ほどのプログラムでできるので，おまけとして説明しましょう．言語はPythonですが，難しいことはしていないので，他の言語に直すのも簡単です．

並びはtrainsという名前の配列（リスト）に1文字ずつ順に入れます．それに対して矢線を引くわけですが，ここではlinesという配列を使って表します．これはtrainsと同じ長さで，最初はすべて0が入っていますが，プログラムを動かした後は，それぞれの区間の矢線の本数が入っているようにします．

ではソースを見てみましょう．コメントもつけました．最初2行は上述の2つの配列の用意で，その次にprintで並びを打ち出しておきます．

```
trains = ['A','D','C','B','E']   # ここでは問[2]の◎のデータを入れてある
lines = [0] * len(trains)        # 0をtrainsの長さと同じだけ並べる
print(trains)                    #  まずtrainsを打ち出しておく
for i in range(0, len(trains) - 1):       # iを列の先頭から最後の1つ前まで
  for j in range(len(trains) - 1, i, -1): # jを最後からiの1つ後まで逆順で調べる
    if trains[j] < trains[i]:             # jの位置の字がiの位置より前なら
      for k in range(i, j + 1):           # iからjまで
        lines[k] = lines[k] + 1           # 矢線の本数を1増やす
      print(lines)                        # linesを打ち出す(分かりやすさのため)
      break                               # i番の処理が終わったのでループ抜け出す
print(max(lines))                # 最後にlinesの最大値を打ち出す
```

外側のforはiをtrainsの先頭から最後の1つ手前まで変えながら，trains[i]から引き出される矢線があるか調べます．最後が除かれているのは，最後はもうそれを追い越す列車はないからです．

内側のforはjをtrainsの最後からiの1つ先まで逆順で変えながら，その次のifでtrains[j]がtrains[i]より小さなjを見つけます．逆順なのは，できるだけ後ろのjを優先して見つけたいからですね．

見つかったら，iからjまで矢線を引き（というのは、ここでは配列linesのiからjまでの値を1増やすことになります），分かりやすいようにlinesを打ち出し，breakで内側ループを抜けて次のiに進みます．

もしtrains[j]がtrains[i]より小さなjが見つからなかったら，そのiは後ろにより大きい文字しかなかったので，矢線は引かないまま次のiに進みます．

このような，内側のforの中にifがあって，見つかったら何かしてbreak，というのはやや高度な定石ですので，覚えておくとよいかと思います．外側のforが終わったら，最後に矢線の重なる最大本数ということで，max(lines)を打ち出して終わります．実行例を見てみましょう．

```
['A', 'D', 'C', 'B', 'E']
[0, 1, 1, 1, 0]
[0, 1, 2, 2, 0]
2
```

確かに，問題[2]の⓪の矢線を引いた図と合っています．いかがですか，この問題はプログラミングは出てこない問題でしたが，このプログラムなら，問題になっていてもおかしくはないと思います．

なおこのプログラムでは，それぞれの位置の矢線の本数を表示するだけなので，たとえば E, D, A, B, C と D, B, A, E, Cは図で描くと異なりますが，ともに[1, 2, 2, 2, 2]と表示されます．

おまけ2：もう少しひねるとどうか？

この問題を発展させる別の方向は何があるでしょうか．問題文の冒頭にはいくつかの条件がコンパクトに書かれていました．

- 十分長い一直線状の線路
- 本線と副線に分岐
- 逆走はできない
- 各列車にはアルファベット1文字でそれぞれ異なる列車名
- 列車同士は十分な間隔をあけて走る
- 本線と副線に分岐した個所は1列車分の長さしかない
- 副線上で列車は停止できるが，本線上では列車は停止せずに通過

これらを変更するのはどうでしょう．線路が限られた長さや環状その他の配線だったり，ある区間に副線が2本（本線と合わせて線路が3本）あったり，走る方向を逆転できたり，同じ名前の列車があったり，列車が結合したり分割できたり（追い越し操作の数が変わる，副線ごとに長さに制限があって列車がおさまらなくなる場合があるなど），分岐個所の長さが2列車ぶんだったり，本線でも停止可能（つまり副線との区別がない）だったり．これらの条件の変更で、いろいろまた別の問題ができそうです．

たとえば，列車が両方向に走れ，分岐が1個所だけだが十分な長さを持つとしたら，何回分岐を使えば昇順の列にできるか，というのは典型的な（この問題よりもよく見かけそうな）問題だと思います．興味があれば問題を作ったり解いたりして見ませんか．

また，プログラミングコンテストなどの問題としてもこの手の問題は定番かと思います．代表例として，ICPCというコンテストの国内予選に出た問題を挙げておきます[3]．

おまけ3：「整列」との比較など

冒頭で言及した「整列」アルゴリズムが，ネタとしてしか出てこないのでは申し訳なさそうなので，最後に「整列」とこの「列車の並べ替え」を比較してみましょう．まず，両方の問題で共通なのは以下です．

- 扱うものは，いくつかの1次元の入れ場所に並んで入る．
- 2つのものの大小は決まった個所で比較される．

そして，「整列」の特徴は次の点です．

- 扱うものは配列に入っていて，どこでも場所を指定して出し入れ可能．

一方，「列車の並べ替え」の特徴は次の点です．

- 扱うものは，線路に端から詰めていく必要がある．

こうして見ると，意外にその差は小さいように思えます．配列の，どこでもアクセスできるという性質は，非常に強力であり，それを利用する整列にはさまざまな（用途に応じた）アルゴリズムがあります．一方で，どこでもアクセスを実現するのは，ハードウェア的にはかなり負担なのです．

そして実は！　メモリに入りきらない大量のデータを整列する方法（外部ソート）としては，「端から詰めていく」アルゴリズムである，マージソートが使われます[4]．これはなんとなく，列車と線路が思い起こされます．興味がある人は調べてみてください．

以上は普通のコンピュータでデータを並べ替える場合でしたが，情報入試委員会の中で原稿を見ていただいているとき，谷聖一先生（日本大学）から，もう少し広い範囲の整列について言及してはとの示唆をいただきました．

ハードウェアよりの方向では，並列性を活かして高速な整列を行うソーティングネットワーク[5]や，ハイパーキューブなど並列計算機での整列アルゴリズムの研究があります．ソーティングネットワークは，以前小学生対象のイベントで，床いっぱいにネットワークを描いて，数字を持った子どもたちが横に並んでネットワークに入っていくと，出口では数字の小さい順に並んでいる，というのを体験してもらいました．

理論に近いほうでは，配列の代わりに連結リストを使うと，「端から詰めていく」方法しか使えないので速さの限界が低くなるなど，いろいろな研究があります．谷先生に教えていただくまで知らなかったのですが，理論的な整列方法にスパゲティソート[6]というアルゴリズムがあります．これは，スパゲティの束を軽く握ってテーブルに立て，一番長いのもの，2番目のもの……と順に取り出していくというもので，スパゲティの本数nに比例した時間で済むという，高速なアルゴリズムです．入力データをスパゲティに変換するのが大変ですが（笑）……理論も楽しいでしょう？

と，ここまで書いておしまいのつもりだったのですが，神藤健朗先生（世田谷学園高等学校）がSNSにこの問題のことを書いていらして，自分の高校で「逆走を許してシェーカーソート[7]の問題に改題した」とのことでした．その設定は，先に提案したものですね！　なお，シェーカーソートというのはバブルソートの改良版で，小→大の向きの比較交換と大→小の向きの比較交換を交互に行っていく整列アルゴリズムです．

そう言われてみると，1個所の副線で往復しながら並べ替えを行うとき，まず最大の列車を止めて一番右に入れ，次に最大から2番目の列車を止めて右から2番目に入れ，のような入れ替えをするとバブルソートそのものですね（実際にはソートアルゴリズムとしては，どれが最大や2番目等かを見出すために副線に「今の所最大」を止めて次々に来る列車と比べることになります）．そして，これをシェーカーソートにするのも簡単です．

こうして見ると，私は第1印象で列車の入れ替えは整列とは違う問題だと思ってしまったのですが，今頃になってバブルソートやシェーカーソートに近い問題だったと分かったわけです．立命館大学の中の人，ごめんなさい．そして気づかせてくださった神藤先生，ありがとうございました，

「並べ替え」の問題はいかがでしたか？

そういうわけで，「列車の並べ替え」の問題はどうでしたか．問題文があまり長くなく，線路と列車というイメージがなじみやすく，各問題も前の問題が（1問目は最初の例の説明が）ヒントになるようにできていて，問題ごとに高度な内容に進むようにできています．しかし，問題量としてはかなりコンパクトです．

さらに，おまけを3つも書いてしまいましたが，プログラムの導入もあり得る，さまざまな違う方式へのつながりもある，整列アルゴリズムとの関係もある，ということでした．そういう発展性の点からも，立命館大学のこの試作問題はなかなか良い問題だと思うのですが，皆様はどうお考えですか．

参考文献

1）立命館大学：2026年度独自入試問題で「情報」を導入 〜学部個別配点方式で「情報型」新設〜，https://www.ritsumei.ac.jp/profile/pressrelease_detail/?id=1102
2）立命館大学：一般選抜 「情報」の本学独自入試の導入について，https://ritsnet.ritsumei.jp/admission/general/Informatics.html
3）ICPC国内予選2006 Problem B 列車の編成パートII，http://www.deqnotes.net/acmicpc/3007/ja
4）ソート，ウィキペディア，https://ja.wikipedia.org/wiki/%E3%82%BD%E3%83%BC%E3%83%88
5）ソーティングネットワーク，ウィキペディア，https://ja.wikipedia.org/wiki/%E3%82%BD%E3%83%BC%E3%83%86%E3%82%A3%E3%83%B3%E3%82%B0%E3%83%8D%E3%83%83%E3%83%88%E3%83%AF%E3%83%BC%E3%82%AF
6）スパゲティソート，ウィキペディア，https://ja.wikipedia.org/wiki/%E3%82%B9%E3%83%91%E3%82%B2%E3%83%86%E3%82%A3%E3%82%BD%E3%83%BC%E3%83%88
7）シェーカーソート，ウィキペディア，https://ja.wikipedia.org/wiki/%E3%82%B7%E3%82%A7%E3%83%BC%E3%82%AB%E3%83%BC%E3%82%BD%E3%83%BC%E3%83%88

（2025年7月2日受付）
（2025年8月1日note公開）

■**久野　靖**（正会員）
電気通信大学 特命教授，筑波大学 名誉教授．理学博士．プログラミング言語，ユーザインターフェース，情報教育に興味を持つ．

情報処理学会ジュニア会員へのお誘い

小中高校生，高専生本科〜専攻科1年，大学学部１〜3年生の皆さんは，情報処理学会に**無料で入会**できます．**会員になると有料記事の閲覧，情報処理を学べるさまざまなイベントにお得に参加できる等のメリットがあります**．ぜひ，入会をご検討ください．入会は**こちら**から！