



Python for bioinformatics



どんぐり研究所 孫 建強

Contents in this document are licensed under [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/).

免責事項

- ここで取り扱う内容は、正確である保証はありません。
- ここで取り扱う内容は、私個人の見解である。私の所属機関を代表する見解ではありません。
- ここで取り扱う内容を利用して、あなたに損害が生じても、一切責任を負いません。
- この内容は、予告なく変更や公開の取り消しを行う場合があります。

コンテンツ

Introduction

1. プログラミング言語概要
2. Python 環境構築

Basic

3. データ型
4. 基本文法

Packages

5. テキスト処理
6. 数値計算 NumPy
7. データ処理 Pandas
8. データ可視化 matplotlib
9. バイオインフォマティクス

Advanced

10. オブジェクト指向

プログラミング言語概要



- プログラミング言語
- データ構造とアルゴリズム

プログラミング言語概要



- プログラミング言語
- データ構造とアルゴリズム

プログラミング言語

特定のタスクを達成させるための一連の操作を、コンピューターに理解可能な形で記述するための言語。

プログラミング言語とゲーム

プログラミング言語をゲームに例えると理解しやすい場合がある。例えば、ゲームではコントローラーを通じて入力し、上に進む、左に進む、右に進む、下に進むなど基本操作（命令）を入力し、キャラクターを制御する。これらの一連の入力（上下左右の順序や回数）がプログラミング言語である。

言語仕様

U	上へ進む
D	下へ進む
L	左へ進む
R	右へ進む
0-9	進むマス数



L4
D3
R3
D3
R3
U3
R3
U3
R3



プログラミング言語

プログラミング言語を学ぶ上で、重要なのは理論的な思考力を身につけることである。ある作業をコンピューターに処理させたいとき、その作業を基本的な操作に分解し、それらを正しい順序で組み合わせ、コンピューターが理解できる形に置き換える必要がある。このような作業の分解や組み合わせなどが大切である。一方で、プログラミングの文法や常用関数の名前や使い方は、あまり重要ではなく、都度に調べればよい。

プログラミング言語

Q1. ゾウを冷蔵庫に入れるにはどうすればよいのか？

▶ データが既に用意されている。方法（アルゴリズム）を書くだけでよい。

1. 冷蔵庫の扉を開ける。
2. ゾウを冷蔵庫に入れる。
3. 冷蔵庫の扉を閉める。

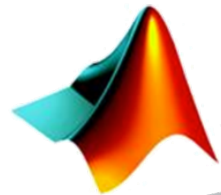
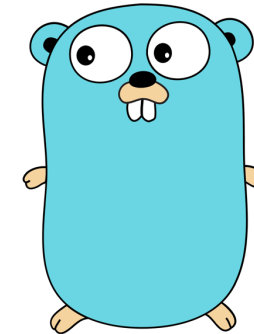
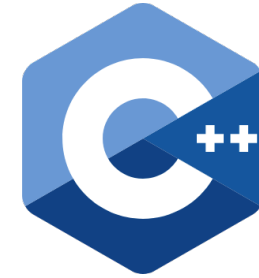
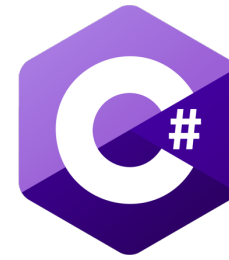
プログラミング言語

Q2. キリンを冷蔵庫に入れるにはどうすればよいのか？

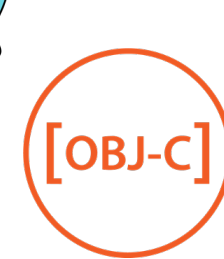
▶ データが既に用意されている。方法（アルゴリズム）を書くだけでよい。

1. 冷蔵庫の扉を開ける。
2. 象を冷蔵庫から出す。
3. キリンを冷蔵庫に入れる。
4. 冷蔵庫の扉を閉める。

プログラミング言語



MATLAB



Swift

現在、数百種類以上のプログラミング言語が使われている。機能に着目すると、様々な目的に使用できる汎用型言語と統計やウェブページ作成などの特定の目的で使用できる専用型言語に分けることができる。これらプログラミング言語同士に優劣はなく、目的に応じて使い分けされている。

1990

2000

2010

🔧 Python 0.9

🔧 Python 1.0

Guido van Rossum が C 言語のように機能が多くて、shell のように簡単に描けるプログラミング言語を開発した。プログラミング言語の名前はBBC コメディ番組『空飛ぶモンティ・パイソン』にちなんで Python と名付けられた。Python は性能重視に作られた言語ではないため、性能を高めたい場合は C 言語でライブラリー .so を作成して Python に読み込むことで対応する。

🔧 Python 2.0

Python の基本的な特徴が決定される。

- オブジェクト指向型
- 動的型付け言語
- 他言語との互換性
- パッケージ

また、2004 年に ウェブ開発フレームワーク Django のリリースにより、Dropbox、YouTube などのウェブに Python が使われるようになる。

🔧 Python 2.7

🔧 Python 3.0

Python 2.x が盛んに使われる中、Python 3.0 がリリースされる。2008 年以降に、2.x と 3.0 系が並行して使われる時代に入る。機械学習、人工知能などパッケージが開発され、Python ユーザー数がさらに増える。現在では 3.x が主流となり、2.x は 2020 年を持ってサポート終了。これから Python を始めるとき 3.7 以降を使うことをお勧めする。

🔧 Python 3.7

1990

2000

2010

Python 0.9

Python 1.0

PIL

Python 2.0

SciPy

matplotlib

Python 2.7

OpenCV

NumPy

Pandas

scikit-learn

Python 3.0

Anaconda

Seaborn

Tensorflow

PyTorch

Keras

Python 3.7

Python

データ処理

視覚化

画像解析

機械学習

プログラミング言語概要

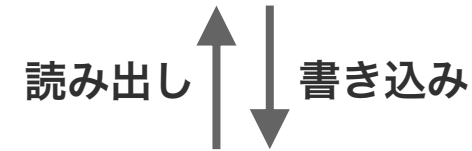


- プログラミング言語
- データ構造とアルゴリズム

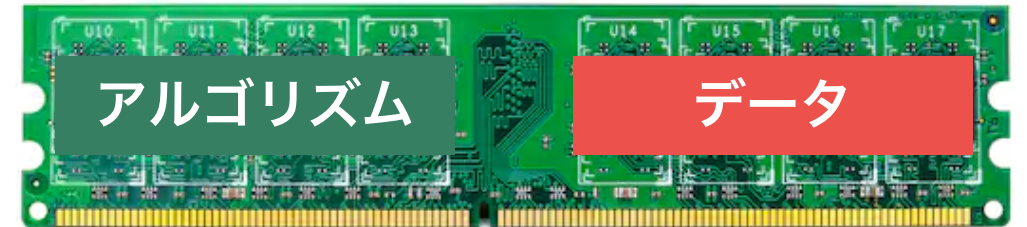
データ構造とアルゴリズム

コンピュータでデータを処理するためには、データや処理手順（アルゴリズム）をコンピュータのメモリに保持させる必要がある。CPU はメモリに保持されたアルゴリズムに従い、指定されたメモリ位置にあるデータを読み込み、計算を行い、その計算結果をメモリの決められた位置に保存する。

CPU

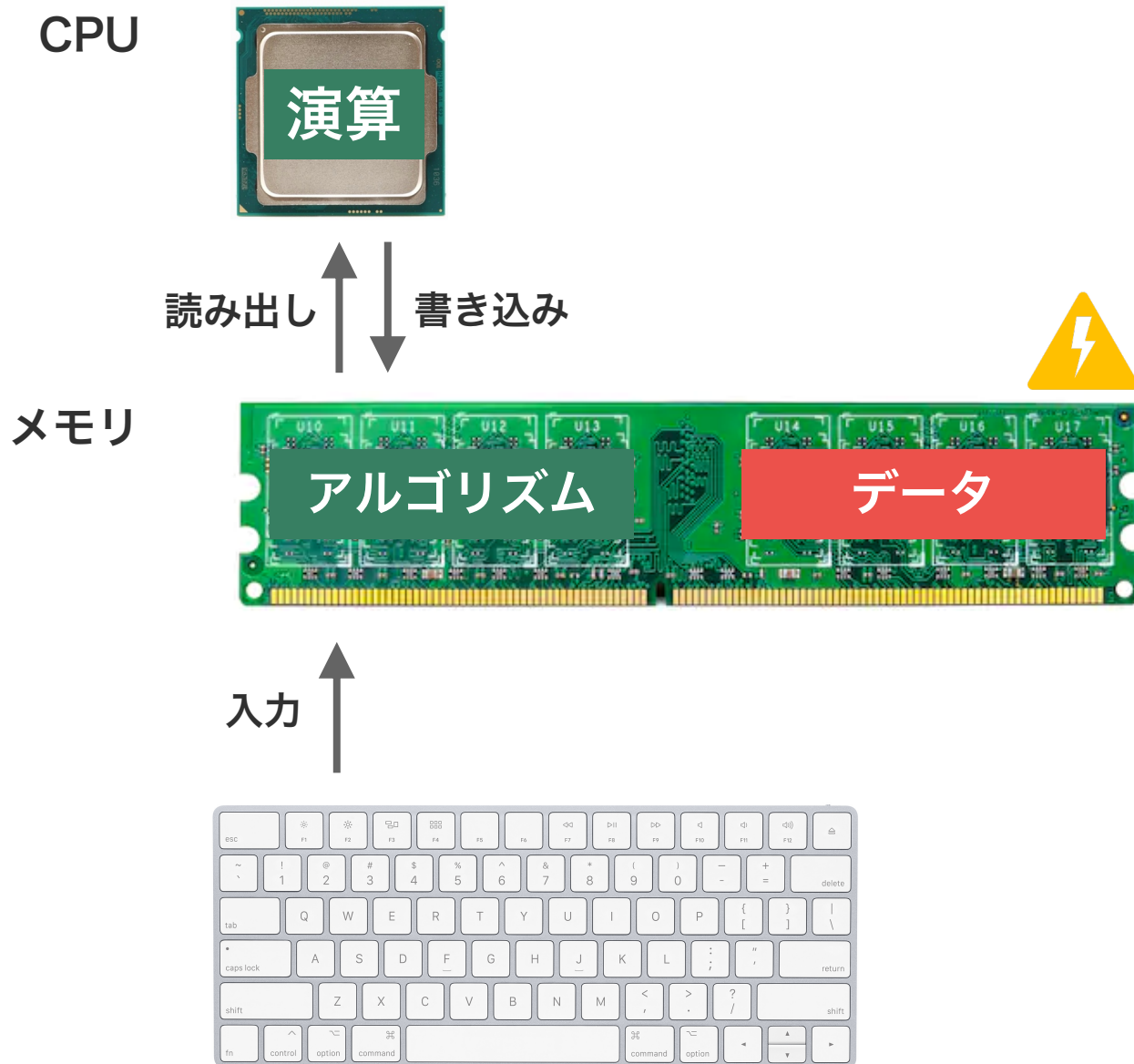


メモリ



データ構造とアルゴリズム

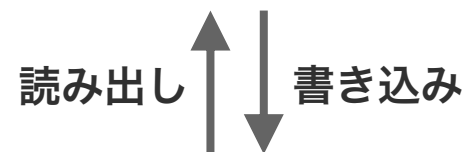
メモリは電気で情報を記録している。コンピュータの電源を切ると、メモリ上の内容は消える。そのため、データを解析する度に、アルゴリズムやデータを直接メモリに書き込む必要がある。この操作は非常に煩雑である。



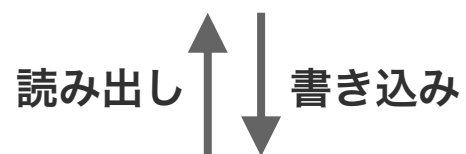
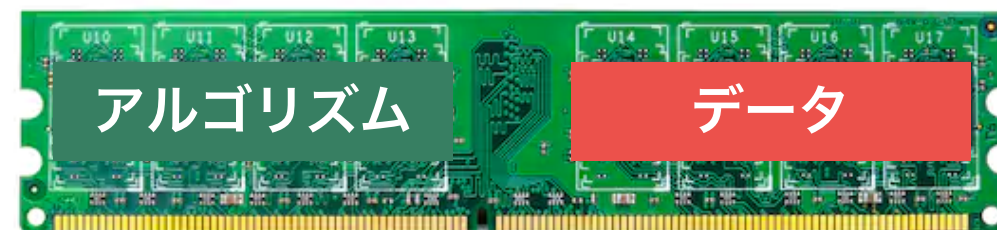
データ構造とアルゴリズム

メモリは電気で情報を記録しているのに対して、ハードディスクは磁気で情報を記録している。ハードディスクは、電気のない状態でも常に情報を保持し続けることができる。このハードディスクを利用すれば効率的なデータ解析が可能になる。その操作として、① アルゴリズムやデータを磁気で情報を記録する装置であるハードディスクに保存し、② 必要に応じてハードディスクからメモリに自動的にコピーしてから解析する。こうすることで、コンピュータを再起動させたときも、すぐに②の操作から再開できる。

CPU



メモリ



HDD

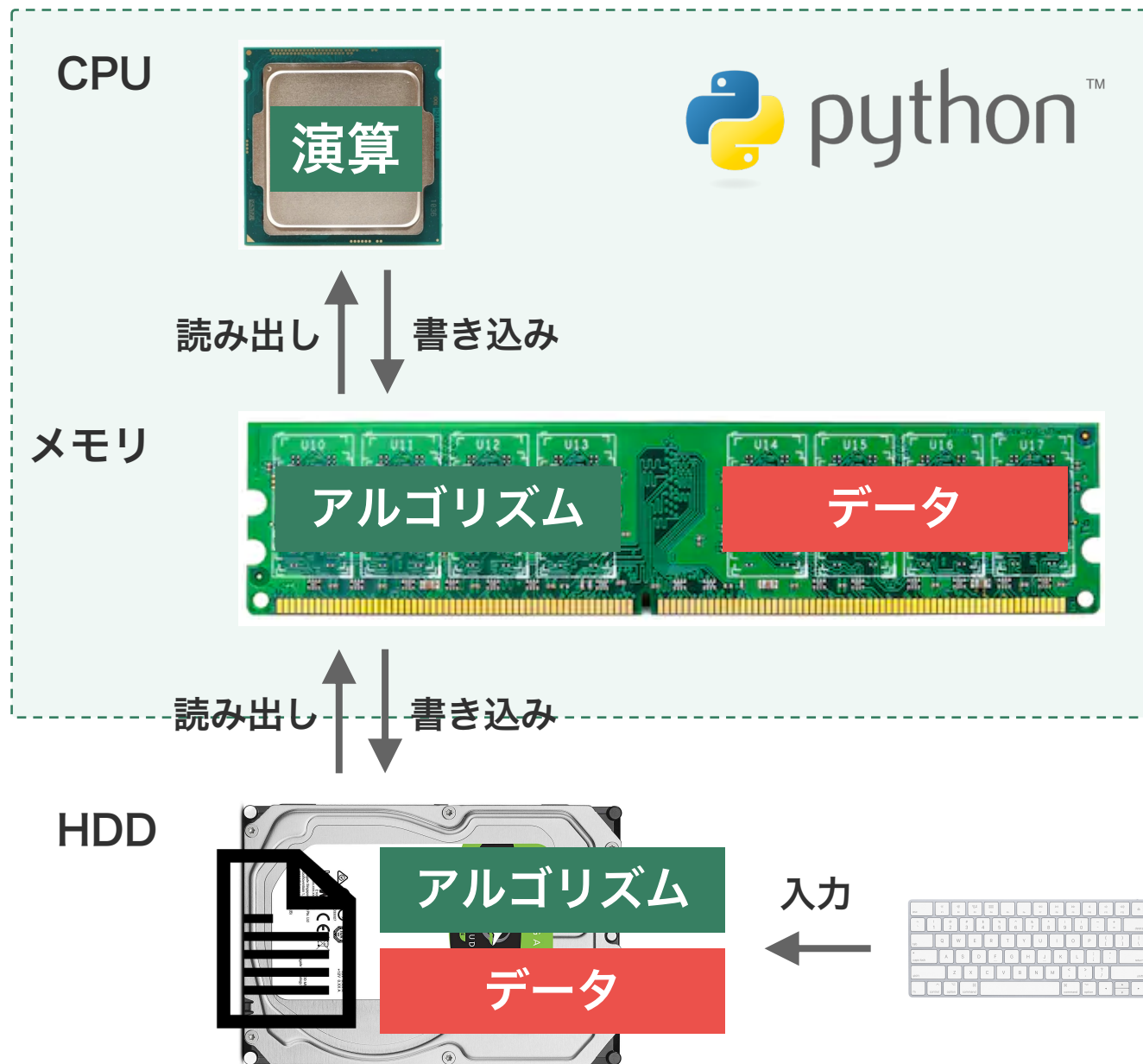


入力



データ構造とアルゴリズム

アルゴリズムおよびデータを Python 文法に従ってファイルに記述すれば、Python エンジンが、そのファイルの情報をメモリ上に展開し、CPU やメモリ間の制御を行いながら演算を行う。



データ構造とアルゴリズム

データ

```
a = 120  
takeout = True  
has_coupon = True
```

アルゴリズム

```
s = 0  
if has_coupon:  
    s = a * 0.95  
  
if takeout:  
    s = a * 1.08  
else:  
    s = a * 1.10
```

```
print(s)
```



データとアルゴリズムを 1 つのファイルに記述し、それを Python に実行させることで、Python がメモリや CPU 間の様々な制御を行う。

データ構造とアルゴリズム

データ



- 整数
- 小数
- 文字
- ベクトル
- 行列



- 変数
- リスト
- ディクショナリ
- 配列
- データフレーム

アルゴリズム



- 探索
- 並べ替え
- 線型計画法
- 待ち行列理論
- 動的計画法



- 条件構文
- 繰り返し構文
- 関数

アルゴリズム – 探索

次の数列の中に 9 が存在するかどうかを調べよ。ただし、コンピュータは、複数の作業を同時に実行できないことに注意せよ。

2	6	4	9	3	8	0	5	1	7
---	---	---	---	---	---	---	---	---	---

アルゴリズム – 探索

次の数列の中に 9 が存在するかどうかを調べよ。ただし、コンピュータは、複数の作業を同時に実行できないことに注意せよ。

2	6	4	9	3	8	0	5	1	7
---	---	---	---	---	---	---	---	---	---

2	6	4	9	3	8	0	5	1	7
---	---	---	---	---	---	---	---	---	---



2	6	4	9	3	8	0	5	1	7
---	---	---	---	---	---	---	---	---	---



2	6	4	9	3	8	0	5	1	7
---	---	---	---	---	---	---	---	---	---



2	6	4	9	3	8	0	5	1	7
---	---	---	---	---	---	---	---	---	---



アルゴリズム – 探索

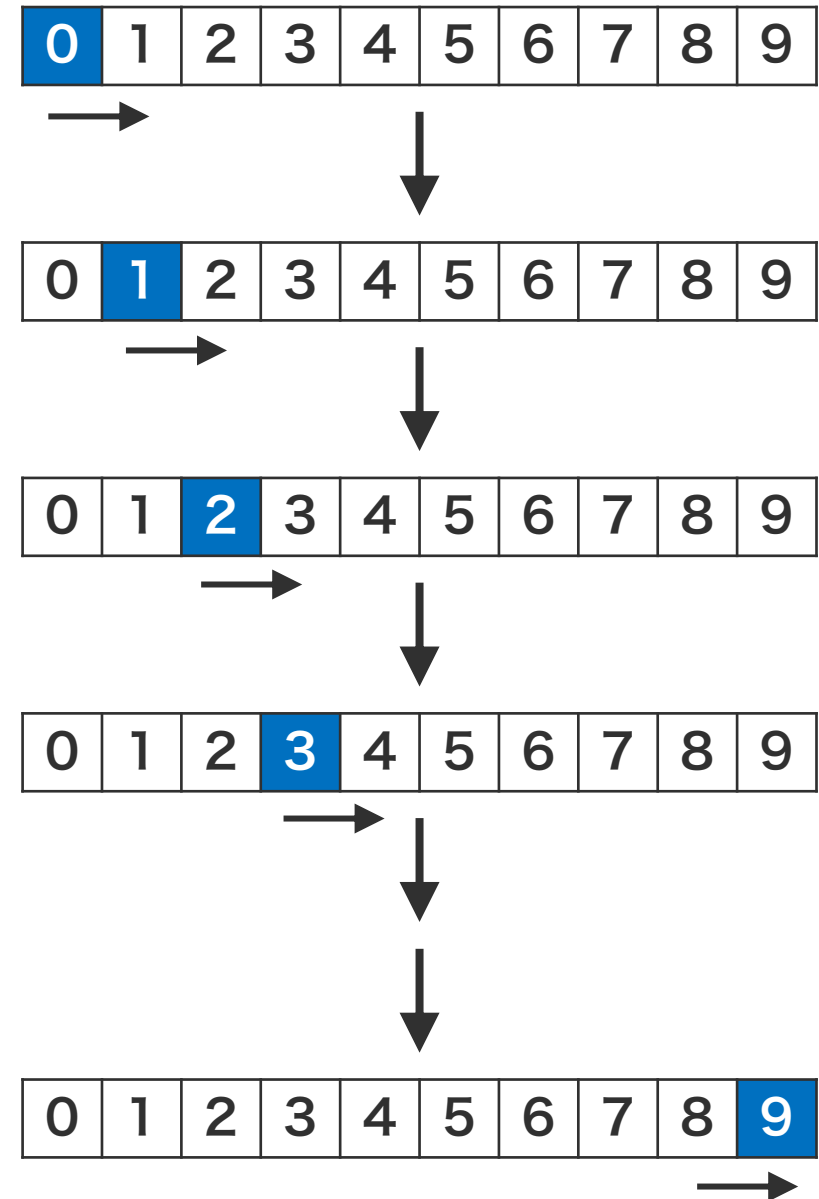
次の数列の中に 9 が存在するかどうかを調べよ。ただし、コンピュータは、複数の作業を同時に実行できないことに注意せよ。

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

アルゴリズム – 探索

次の数列の中に 9 が存在するかどうかを調べよ。ただし、コンピュータは、複数の作業を同時に実行できないことに注意せよ。

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



アルゴリズム – 探索

次の数列の中に 9 が存在するかどうかを調べよ。ただし、コンピュータは、複数の作業を同時に実行できないことに注意せよ。

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



アルゴリズム – 並べ替え

次の数列の各要素を昇順に並べ替えよ。ただし、コンピュータは、複数の作業を同時に実行できないことに注意せよ。

6	3	4	1	2	0	9	5	8	7
---	---	---	---	---	---	---	---	---	---



0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

アルゴリズム – 並べ替え

次の数列の各要素を昇順に並べ替えよ。ただし、コンピュータは、複数の作業を同時に実行できないことに注意せよ。

6	3	4	1	2	0	9	5	8	7
---	---	---	---	---	---	---	---	---	---



0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

6	3	7	1	2	0	9	5	8	4
---	---	---	---	---	---	---	---	---	---



- 1) 1 番目の要素と 2 番目の要素を比較する。
- 2) 2 番目の要素が大きければ、何もしない。
- 3) 1 番目の要素が大きければ、
 - i. 2 番目の要素を変数 tmp に移動する。
 - ii. 2 番目の位置に空きができるので、1 番目の要素をそこに移動する。
 - iii. 1 番目の位置に変数 tmp の値を移動する。



3	6	7	1	2	0	9	5	8	4
---	---	---	---	---	---	---	---	---	---

アルゴリズム – 並べ替え

次の数列の各要素を昇順に並べ替えよ。ただし、コンピュータは、複数の作業を同時に実行できないことに注意せよ。

6	3	4	1	2	0	9	5	8	7
---	---	---	---	---	---	---	---	---	---



0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

6	3	7	1	2	0	9	5	8	4
---	---	---	---	---	---	---	---	---	---



3	6	7	1	2	0	9	5	8	4
---	---	---	---	---	---	---	---	---	---



3	6	7	1	2	0	9	5	8	4
---	---	---	---	---	---	---	---	---	---



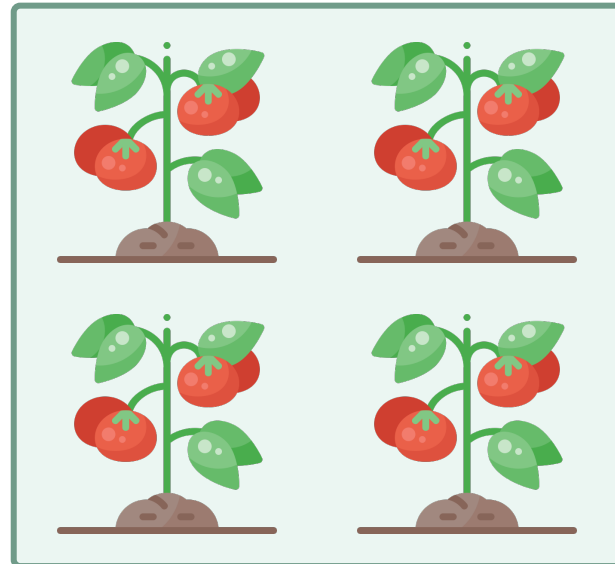
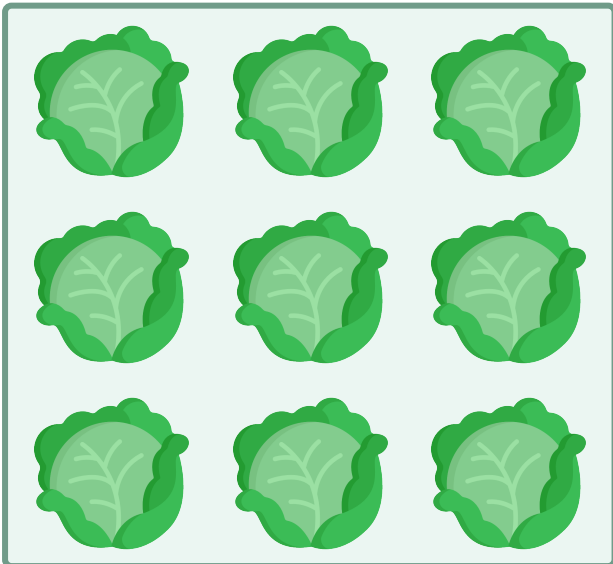
3	6	1	7	2	0	9	5	8	4
---	---	---	---	---	---	---	---	---	---



0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

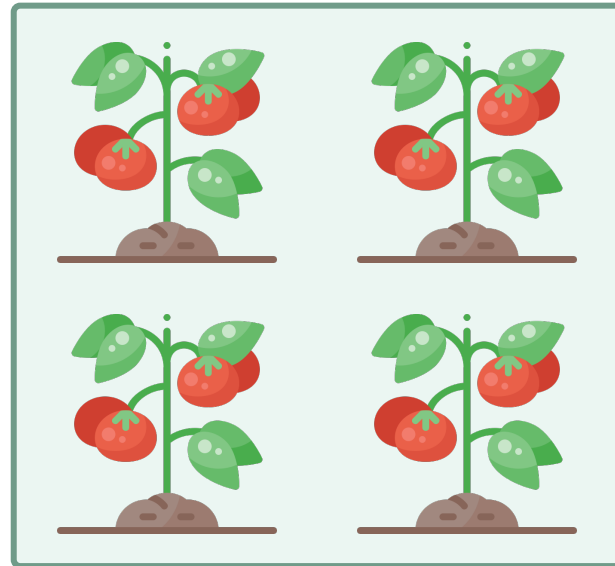
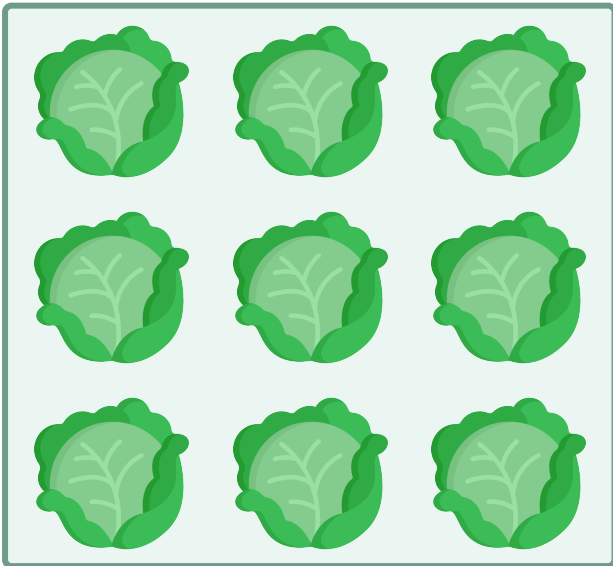
アルゴリズム – 線形計画法

1 m² にキャベツなら 9 株、トマトなら 4 株植えることができる。キャベツ 1 株 100 円、トマト 1 株 500 円の収入が見込まれる。農作物が育つには、キャベツ 1 株あたり 10 g、トマト 1 株あたり 40g の肥料が必要とされる。あなたは 10 m² の農地と 1kg の肥料を所有している。収入を最大化するためには、キャベツとトマトをそれぞれ何株ずつ植えればよいか。



アルゴリズム – 線形計画法

1 m² にキャベツなら 9 株、トマトなら 4 株植えることができる。キャベツ 1 株 100 円、トマト 1 株 500 円の収入が見込まれる。農作物が育つには、キャベツ 1 株あたり 10 g、トマト 1 株あたり 40g の肥料が必要とされる。あなたは 10 m² の農地と 1kg の肥料を所有している。収入を最大化するためには、キャベツとトマトをそれぞれ何株ずつ植えればよいか。



収入最大化

$$y = 100c + 500t$$

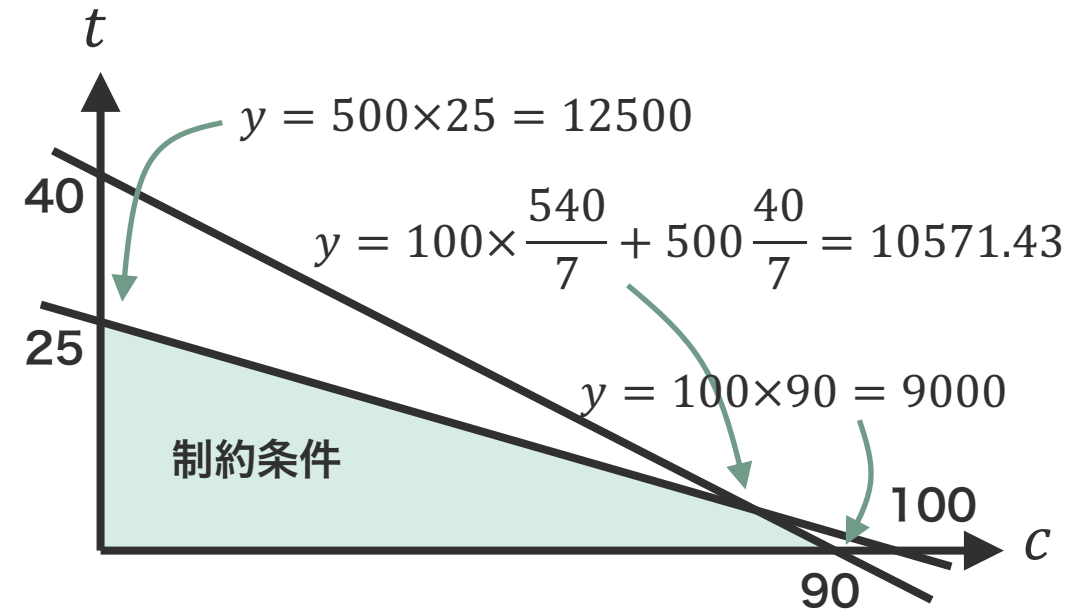
制約条件

$$\frac{1}{9}c + \frac{1}{4}t \leq 10$$

$$10c + 40t \leq 1000$$

$$c \geq 0$$

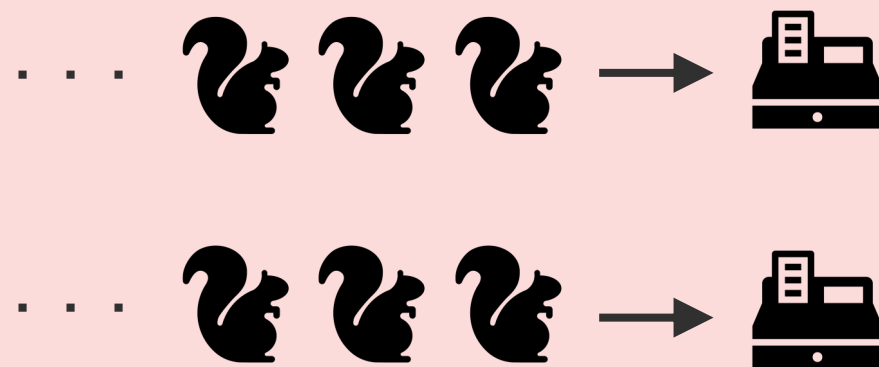
$$t \geq 0$$



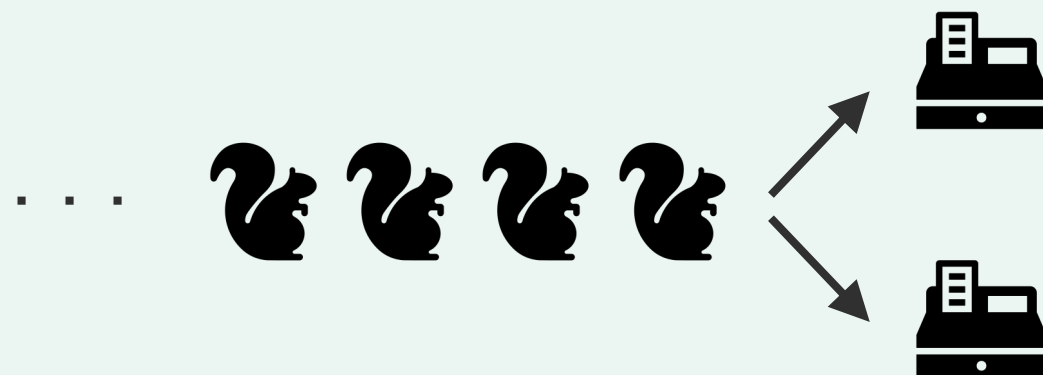
アルゴリズム – 待ち行列理論

店内には無数の買い物客がレジ前で会計待ちをしている。それぞれの客が購入する物品の量はランダムである。カゴ 1 つ分の量もあれば、ペットボトル 1 本の量もある。店内で稼働しているレジが 2 つだけである。レジを活用する方法は次の方法 A と方法 B がある。買い物客の待ち時間を最小化するためには、あなたは方法 A と方法 B のどれを選ぶか。

方法 A

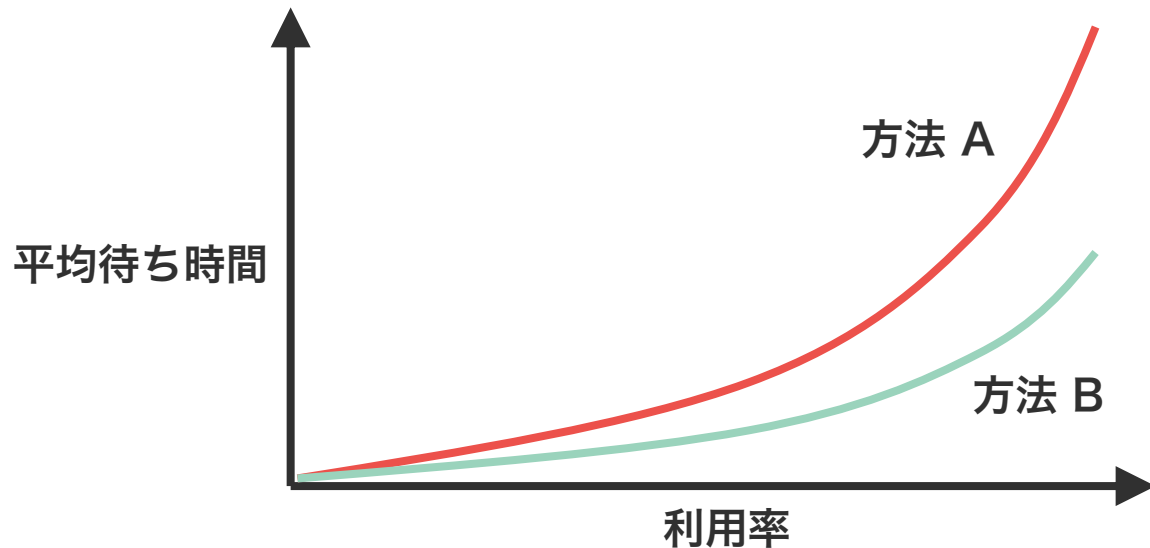


方法 B

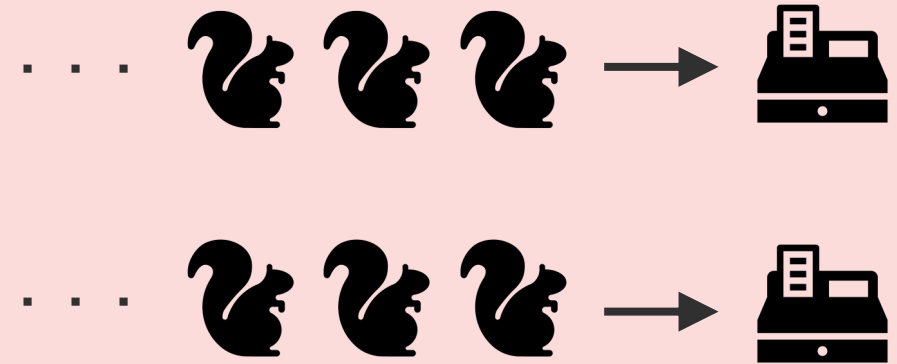


アルゴリズム – 待ち行列理論

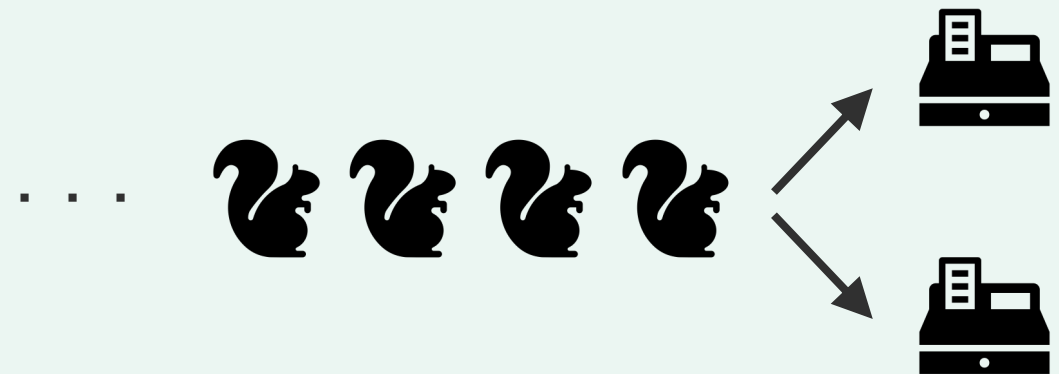
店内には無数の買い物客がレジ前で会計待ちをしている。それぞれの客が購入する物品の量はランダムである。カゴ 1 つ分の量もあれば、ペットボトル 1 本の量もある。店内で稼働しているレジが 2 つだけである。レジを活用する方法は次の方法 A と方法 B がある。買い物客の待ち時間を最小化するためには、あなたは方法 A と方法 B のどれを選ぶか。



方法 A



方法 B



アルゴリズム – 動的計画法

仕事の帰り、最寄駅から家までの間に、次の用事を済ませる必要がある。

- ・ スーパーで買い物する
- ・ ポストで手紙を出す
- ・ 幼稚園で子供を迎えに行く
- ・ 薬局で風邪薬を買う

なるべく負担の少なくするためには、どのような道順で行けば良いのか。ただし、100 m 歩くあたりの負担スコアは、以下のようなものとする。

- ・ 購入物： 100
- ・ 手紙： 1
- ・ 子供： 500
- ・ 風邪薬： 5

