
プログラミング演習2 (1)

イントロダクション

中村、小林、橋本、辻野

まずは

- 教員・TAからの自己紹介を

プログラミング

- ある目的を達成するプログラムの書き方は山のように存在している
 - ある積み木セットを利用して、4つの塔がある城を作れという課題が与えられたときに同じものができらるだろうか？
- また、プログラミングは文章を書くことと似ているため、そこには個性が現れる
 - 文章が一致することはとても少ない

コピーとは？

- 他人から完成したプログラム自体を受け取り、それに何らかの手を加えること提出
- 他人から完成したプログラムの写真を受け取って、それを写して手を加えること提出
- 他人に完成したプログラムをディスプレイに表示し続けてもらい、それを写して手を加え提出

提供しないように！

- 良かれと思ってやりたくなる気持ちはわかります
- 教えるのが面倒でこれみとけって気分になることもあるでしょう
 - でも、同一レポート（課題）の提出があった場合、提供された側が正直に申告しない場合は提供元・提供先の両方が処罰の対象となります
 - そんなしょうもないことで人生を狂わせてはもったいないので、くれぐれも注意してください
- 他の講義でも問題になりつつあります。絶対に課題はコピーさせないこと！

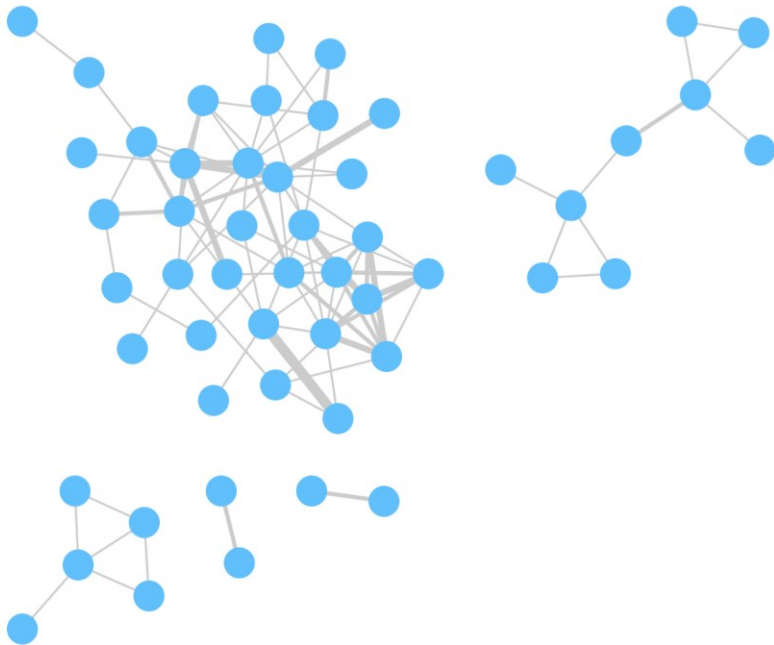
これまで来た質問

- 教えたら類似するのでは？
 - 多少は類似するでしょう。ただ、積み木でお城を作るという課題で同じ形、同じ色の積み木を同じように積むことはあり得るでしょうか？
 - 見せなければ一致しない
- 教えることを禁止するのか？
 - ぜひとも教えて下さい、ただコピーが発生するような行為は慎んでください
- 他人のプログラムを見てはだめなのか？
 - 教えるときは見ざるを得ませんので、その時は見て下さい。ただ、見せて教えないで下さい。

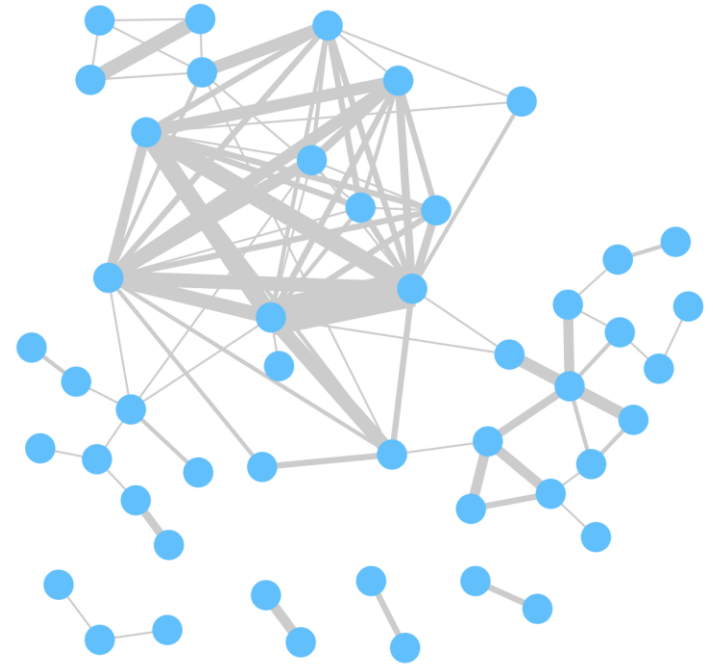
あれだけ言ったのに...

- プログラムの類似度（レーベンシュタイン距離）をもとにしてネットワークグラフを構築したところ、下図のような結果に
 - 線の太さは類似回数を意味しています

発展課題のみ



基本課題 + 発展課題

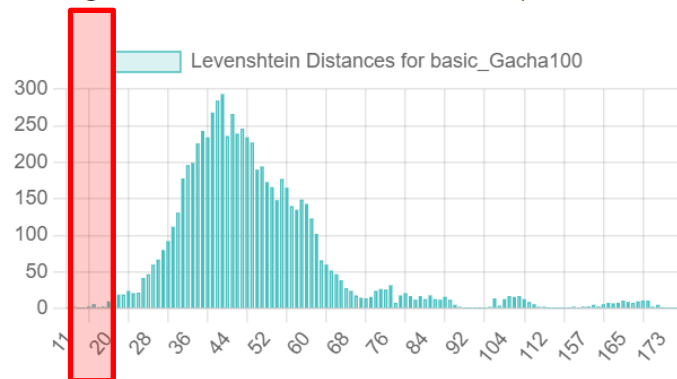


レーベンシュタイン距離

- あるテキストを、どの程度書き換えたかの判定に利用されるもの
 - 通常、それぞれが独立にコードを書いている場合は似通わない
 - 誰かのものをコピーして書き換えた場合は、かなり似通った結果になる

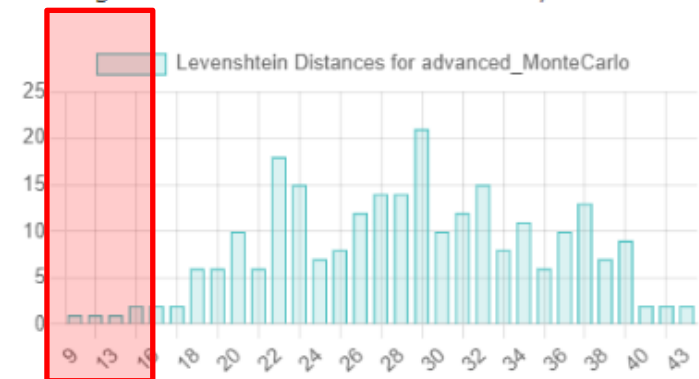
Assignment: basic_Gacha100

Average Levenshtein Distance: 509.30, Standard Deviation: 204.97



Assignment: advanced_MonteCarlo

Average Levenshtein Distance: 288.49, Standard Deviation: 65.40



書き換えの稚拙さ

- 改行位置や変数名変更、順番の変更程度の編集距離はかなり小さい（教えるだけでこんなに似ることはありません）

```
6 - double calcPI(int N)
7 - {
8 -     double y=0;
9 -     for(int k=0;k<=N;k=k+1)
10 -    {
11 -        y=y+pow(-1,k)/(2*k+1);
12    }
13 -    return y*4;
14 }
```

```
16 - void setup()
17 - {
18 -     int a;
19 -     for(a=10;a<=10000000;a=a*10)
20 -     {
21 -         println("N="+a+" "+calcPI(a));
22 -     }
23 }
```

```
5 + double calcPI(int N){
6 +     double a=0;
7 +     for(int i=0;i<=N;i++){
8 +         a=a+pow(-1,i)/(2*i+1);
9     }
10 +    return 4*a;
11 }
```

```
13 + void setup(){
14 +     for(int i=10;i<=10000000;i=i*10)
15 +         println("N="+i+" "+calcPI(i));
16 }
```

類似

```
20 - void itemGacha(){
21 -   for(int i=0;i<50;i++)
22     {
23     items[(int)random(100)]++;
24   }
25 - }
26
27 - void mousePressed() {
28 -
29 -   if(dist(mouseX,mouseY,60,60)<=40)
30     {
31     -   itemGacha();
32     -
33     -   int a=width/100;
34     -   for(int i=0;i<100;i++)
35     -   {
36     -     fill(0);
37     -     rect(i*a,height,a,-items[i]);
38     -   }
```

```
21 + void itemGacha()
22 + {
23 +   for(int i=0;i<50;i++)
24     {
25     items[(int)random(100)]++;
26 +   }
27 + }
28
29 +
30 +
31 + void mousePressed()
32 + {
33 +   if(dist(mouseX,mouseY,60,60)<=40)
34 +   {
35 +     itemGacha();
36 +     int w=width/100;
37 +     for(int i=0; i<100; i++)
38     {
39 +     fill(0);
40 +     rect(i*w,height,w,-items[i]);
```

The 稚拙

5	- int []customer=new int[264];		
6	- int a=0;		
7		4	
		5	+ int[] people=new int[264];
		6	+ int count=0;
		7	+
8	void setup()	8	void setup()
9	{	9	{
10	for(int i=0;i<100000;i++)	10	for(int i=0;i<100000;i++)
11	{	11	{
12	canSit();	12	canSit();
13	}	13	}
14	- float pos=(float)a/100000*100;	14	+ float result=(float)count/100000;
15	- println(pos,"%");	15	+ println(result,"%");
16	}	16	}
17		17	
18	-		
19	void canSit()	18	void canSit()
20	{	19	{
21	for(int i=0;i<264;i++)	20	for(int i=0;i<264;i++)
22	{	21	{
23	- customer[i]=i;	22	+ people[i]=i;
24	}	23	}
25	- int cF=(int)random(0,264);	24	+ int a=(int)random(0,264);
26	- customer[cF]=-1;	25	+ people[a]=-1;
27	- for(int i=1;i<263;i++)	26	+ for(int i=1;i<263;i++)
28	{	27	{
29	- if(customer[i]!=-1)	28	+ if(people[i]!=-1)
30	{	29	{
31	- customer[i]=-1;	30	+ people[i]=-1;
		31	+
32	- }else	32	+ else
33	{	33	{
34	- int random=(int)random(0,264);	34	+ int b=(int)random(0,264);
35	- while(customer[random]==-1)	35	+ while(people[b]==-1)
36	{	36	{
37	- random=(int)random(0,264);	37	+ b=(int)random(0,264);

書き換えたつもり？

- この程度でも当然類似判定されます

4	- float a=0,b=0;	4	
5		5	+ float a=0;
		6	+ float b=0;
		7	+
6	void setup()	8	void setup()
7	{	9	{
8	size(800, 800);	10	size(800, 800);
9	background(255);	11	background(255);
	@@ -11,24 +13,25 @@		
11	}	13	}
12		14	
13	void draw()	15	void draw()
14	{	16	{
15	- int x=(int)random(width);	17	+ float x=random(width);
16	- int y=(int)random(height);	18	+ float y=random(height);
17	-		
18	strokeWeight(1);	19	strokeWeight(1);
19	- if (dist(400, 400, x, y)<400)	20	+ if (dist(400, 400, x, y)<400)
20	{	21	{
		22	+ stroke(255, 0, 0);
21	fill(255, 0, 0);	23	fill(255, 0, 0);
22	- stroke(255,0,0);		
23	a=a+1;	24	a=a+1;
24	-	25	+ }
25	- } else	26	+ else
26	{	27	{

他の人にも転送されます

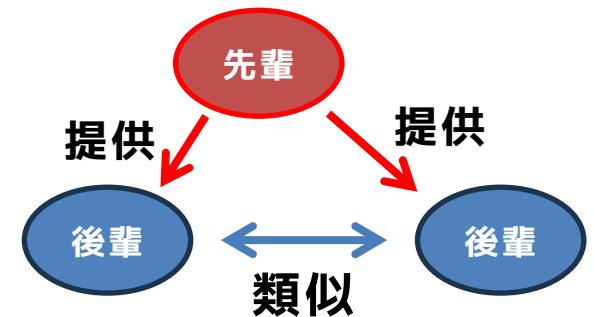
- 成績S → 成績B → 成績C/F という転送経路を確認（コードの書き換え距離から）
 - 今回は甘々判定で、Sの人は減点していませんが
今後は同罪とします

課題提出の怖さ

- 手書きとは異なり、デジタルでの課題提出は、後で何度も類似判定ができてしまう
 - 蒸し返される危険性があることに注意！
- 絶対にコピーしない！ コピーさせない！
ことを注意して下さい
- プログラミング演習2では、プログラミング演習1での類似も踏まえて、類似判定を行います。1で引っかけからなかったからと油断しないように！

先輩にもらわないこと！

- 先輩のプログラムをもらって再利用した場合、そのプログラムを誰が同じようにもらってるかがわかりません
- 先輩のプログラムを使うことで、知らない他者と似通うということになってしまいます！



- 剽窃判定されてしまうため、絶対に先輩のプログラムはコピーしないこと！

ChatGPTについて

- ChatGPTを使って学習することは問題ありませんが、ChatGPTを使って課題を解き提出することは禁止していますので注意
 - カモつきませんしいいことはありません
- 一番危険なのは、**GPTを使うことで、他者とコードが類似してしまう**こと（これで減点されている人もいるかも知れません）
- P演習1では、一致を除くと明らかにChatGPTを使ったと思われるもの以外は減点していませんが、P演習2では厳正に処置します

GPT類似と思われる例

- これくらい違っても類似ペアとして類似判定されます

6	- void setup()	5	+ void setup() {
7	- {		
8	- size(400, 200);		
9	- textSize(16);		
10	- textAlign(CENTER);		
11	- int[] N_values = {10, 100, 1000, 10000, 100000};	6	+ int[] testValues = {10, 100, 1000, 10000, 100000};
		7	+
12	- for (int N : N_values)	8	+ for (int N : testValues) {
13	- {		
14	- double result = calcPI(N);	9	+ double pi = calcPI(N);
15	- println("N = " + N + " " + result);	10	+ println("N = " + N + ": " + pi);
16	}	11	}
17	}	12	}
18	- void draw()	13	+
19	- {		
20	- }		
21	- double calcPI(int N)	14	+ double calcPI(int N) {
22	- {		
23	- double pi = 0;	15	+ double sum = 0.0;
24	- int sign = 1;		
25	- for (int k = 0; k < N; k++)	16	+ for (int k = 0; k <= N; ++k) {
26	- {		
27	- int denominator = 2 * k + 1;		
28	- pi += sign * 1.0 / denominator;	17	+ sum += (pow((-1),k)) / (2 * k + 1);
29	- sign *= -1;		
30	}	18	}
31	- return 4 * pi;	19	+ return sum * 4;
32	}	20	}

次回は厳しく判定します

- 今回減点していないひとについても、記録には残しています
- プログラミング演習2で同等のコピーが発見された場合は、単位なしの判定をします

提供してもらわないように！

- みなさんがコピーしてしまうことにより、コピー提供者も迷惑を被ります
- この講義の後も
 - メディアプログラミング実習（2年春）
 - アルゴリズム基礎（2年春）
 - Webプログラミング実習（2年秋）があるため、逃げたらずっと苦しむだけ。勉強にもなりませんのでしんどいなりに立ち向かいましょう！
- これまで英語や数学とかを勉強してきたときどうしています？ どれくらいの時間をかけてきました？

改めて注意点

- 他人にプログラムを提供しないこと
（書き換えは稚拙で、他者に波及します）
- 他人のプログラムをもらわないこと
（迷惑をかける事になります）
- 先輩のプログラムをもらわないこと
（知らないうちに他人と似ます）
- GPTに課題のプログラムを書かせないこと
（知らないうちに他人と似ます）

さて

- EP演習の発表会どうでしたか？
- 知人の作ったプログラム、発表はどうでしたか？

EP演習の発表会

- 恥ずかしがらずに是非自分が作ったものは
どんどん発信していきましょう！
- 経験上...
 - 完璧なもの・理想的なものができるとまで発信しないひとは、いつまでも発信せず、そして成長していきません
 - 十分でなくてもいいので今あるものを発信できるひとは勢いよく成長していきます
- 作ったものを録画し、研究室配属の面接で見せることができるようにしておこう！

前にも言いましたが、プログラミング習得

- 習得する方法には主に、2つの道があります

(1) 作りたい欲にまかせて、自分で調べつつ色々なプログラムを作っていく

- 本当はこちらが上達速度も早くおすすめです
- ただ誰もがができる道ではないとはわかっています

(2) ひたすら写経していく

- 作りたい欲がない場合、一番手っ取り早いのが勘所を掴むため写経するということになります。ひたすら書いて覚える！
- そのためにプログラミング演習1では `typing.run` を用意していました

プログラミングの習慣をつけよう

- #dailycoding
 - Twitter に毎日作ったプログラムをアップしていくという活動
 - とにかく、毎日作るということを心がけていくこと、ただそれだけ



takawo shunsuke
@takawo

...

気がつくと #dailycoding のハッシュタグを使う人が増えている？「続かない」と思ったり躊躇してる人がいたら、いつ始めても、止めても、再開してもいいのがデイリーコーディングなので、気軽に気負わずに作ったものをシェアしていきましょう👏👏

twitter.com/hashtag/dailyc...

どこでやる？

- OpenProcessingを使っていきましょう！
 - <https://openprocessing.org/>
 - そこでプログラムを書けて、かつURLで共有できちゃいますので、Twitterに投稿しよう！
- Processingでも書けますし、p5.jsでも書くことができます！

p5.js って何？

- Processing と同等の表現を JavaScript で記述可能とするライブラリ
 - 気軽にWebのかっこいいインタフェースを作ることができます
- どれくらい違うのか？
 - 変数に型がなくなります（letやvarで定義）
 - void/int/floatがなくなります
 - 関数の定義はfunctionです
 - sizeの代わりにcreateCanvasです
 - それ以外はほぼ同じです
 - [typing.run](#)のp5.js版を参照

ちなみに

- dailycoding 活動をされている takawo さんがNFTアートとして10000個を販売し、2時間で完売（売上600万ドル以上）
 - ちなみに、売上はProcessingを中心としたコミュニティへ全額寄付
 - <https://generativemasks.on.fleek.co/#/>



繰り返しになるけど

- 「作りたい欲にまかせて、自分で調べつつ色んなプログラムを作っていく」という入り口にみなさんすでに立ててるとおもいます
- この方法は、上達速度も早くおすすめですので、ぜひこの方法で力を伸ばしていける人が増えてほしいと思っています

さてさて

- とはいえ講義です
- プログラミング演習2もまた、その表現のための基礎を作り、そして応用するためにあります

プログラミング演習2について

- プログラミング演習1と同じところ
 - Processingをやります
- プログラミング演習1との違い
 - 予習より復習を重視して下さい
 - typing.runはありません！
 - 中間試験があります（10月28日を予定）
 - 宿題があります
 - グループ制作 / 発表してもらいます
 - 再履修の学生は時間が合わないことも多いためグループ制作か最終課題かを選択可能とします。

グループ制作 / 発表

- 研究室をベースとしてグループ分け
- 他者とプログラムする事の難しさ、楽しさを知り、1人ではできない事を実現する！
 - ぶっちゃけグループ制作は大変です

講義の実施形態

- 9～10月は「クラス（オブジェクト指向）」など座学中心
- 中間試験は10月28日に対面で実施予定
- 11～12月是对面で応用実習系を実施予定
 - ネットワーク、ファイル入出力、XMLやJSON、Web API、フィジカルコンピューティングなど
- 1月に発表会
 - 動画発表＋各自コメント投稿で実施するか、ホールを利用して発表会形式で実施するかは未定
 - ホールを利用すると盛り上がるけど残らないのが難点で動画発表は盛り上がり欠けるけどコメントが残りアーカイブ可能なのがメリット

講義の流れ（あくまで目安）

- 1コマ目：
 - 0～20：宿題の解説
 - 20～50：今日のトピックの解説＋課題の提示
 - 50～100：今日の課題に取り組む（要提出）

- 2コマ目：
 - 0～50：今日の課題に取り組む
 - 50～100：今日の課題の解説（答え合わせ）
 - 次回までの宿題の提示

評価

- 宿題： 20点
- 授業中課題： 20点（要提出）
- 中間テスト： 40点
- 発表会： 20点
- 注意
 - シラバスに小テストの勉強をと書かれてましたが小テストはありません（消し忘れてごめん）
 - 5回休み = 評価対象外
 - PP-Checkerが優秀なため、中間テストまでは課題チェック状況を開示する予定ですが、採点が間に合わないことが判明したら、課題のチェック状況は開示しないよう変更していきます

講義予定（案）

- 09/23 第01回 ガイダンス + 前期の復習 + α
- 09/30 第02回 クラス
- 10/07 第03回 クラスと継承
- 10/14 第04回 クラスと継承と色々
- 10/21 第05回 復習
- 10/28 第06回 中間テスト（40点）
- 11/11 第07回 振り返りとファイル入出力
- 11/18 第08回 ネットワーク
- 11/25 第09回 Web API
- 12/02 第10回 フィジカルコンピューティング
- 12/09 第11回 フィジカルコンピューティング
- 12/16 第12回 フィジカルコンピューティング
- 12/23 第13回 色々
- 01/20 第14回 発表会

宿題の提出方法

- 宿題の提出期限は次回の講義開始30分前
(13:00まで)
 - それ以降のものは受理されませんので早め早めに提出をよろしくです！

- PP-Checkerに提出

本日の課題

- 参加開始
 - まずは参加開始（出席確認）申請
- String型について
- 課題提示

文字列型

- String 型は、文字列を扱うためのクラス
 - 「中村聡史」「明治大学 総合数理学部」
- 文字列を扱うにはこういった機能が必要？
 - 文字列の長さを取得する
 - 文字列にある文字が含まれているかを調べる
 - 文字列を部分的に置き換える
 - 文字列が一致しているか調べる
 - n文字目の文字を取得する
 - などなど

例

- 塩基配列の特徴を分析したい
 - Aはアデニン、Tはチミン、Gはグアニン、Cはシトシンで、この並びや変異が意味を持つ
 - A、T、G、Cの比率や、2連続や3連続、4連続する塩基の比率がその特徴を表していたりする

[演習] 塩基配列を扱ってみよう

- DNA塩基配列を文字列処理してみよう！
 - 塩基配列は、ATGCの4文字からなるもの
 - 下記をダウンロードしよう！
 - test_Covid19Genome.pde
 - SARS-CoV-2（新型コロナウイルス）の塩基配列
- 演習
 - 塩基配列の長さを出力しよう
 - 最初の10文字を出力しよう
 - 最初にAが4つ連続しているのは何文字目か？
 - <https://www.ncbi.nlm.nih.gov/nucore/MN908947>

Stringクラスのメソッド

- `charAt(num)`; `num`文字目の文字を返す（0から始まる）
- `indexOf(文字列)`; 入力された文字列が何文字目か？
- `indexOf(文字列, n)`; `n`文字目以降で入力された文字列が何文字目か？
- `length()`; 入力された文字の文字数を返す
- `substring(x)`; `x`文字目から最後までを出力
- `substring(x, y)`; `x`文字目から`y-1`文字目までを出力
- `toLowerCase()`; 全てを小文字に変換する
- `toUpperCase()`; 全てを大文字に変換する
- `replace(文字列A, 文字列B)`;
 - 文字列Aを文字列Bに変更する
- `split(文字列)`; 文字列を分割

<http://processing.org/reference/String.html>

<http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>

Stringクラスのメソッド

- `charAt(num)`; num文字目の文字を返す（0から始まる）
- `indexOf(文字列)`; 入力された文字列が何文字目か？
- `indexOf(文字列, n)`; n文字目以降で入力された文字列が何文字目か？
- `length()`; 入力された文字の文字数を返す
- `substring(x)`; x文字目から最後までを出力
- `substring(x, y)`; x文字目からy-1文字目までを出力
- `toLowerCase()`; 全てを小文字に変換する
- `toUpperCase()`; 全てを大文字に変換する
- `replace(文字列A, 文字列B)`;
 - 文字列Aを文字列Bに変更する
- `split(文字列)`; 文字列を分割

<http://processing.org/reference/String.html>

<http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>

塩基配列

```
test_Covid19Genome.pde
```

```
// https://www.natureasia.com/ja-jp/nature/pr-highlights/13205
```

```
// https://www.ncbi.nlm.nih.gov/nuccore/MN908947
```

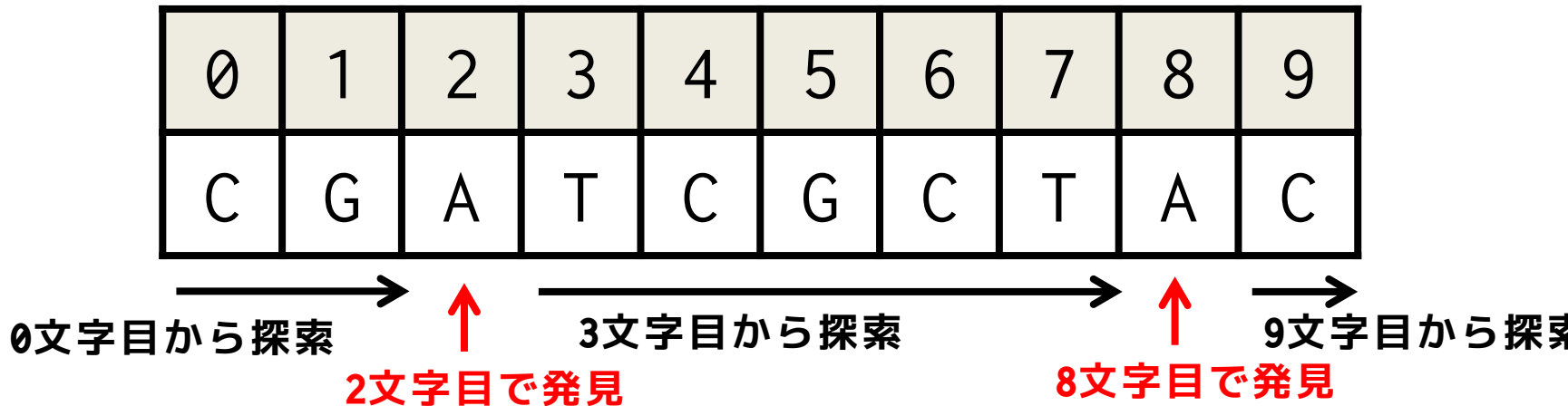
```
// SARS-CoV-2の塩基配列
```

```
String strGENOME = "ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATC...
```

- 塩基配列の長さは？
 - `.length()` を利用する
- 最初の10文字を出力しよう
 - `.substring()` を利用する
- 最初にAが4つ連続しているのは何文字目？
 - `.indexOf()` を利用する

[演習] 続き

- 「A」の出現回数を求めて出現回数を表示しよう！
- 戦略
 - charAtで1文字ずつ見て行って、その文字が「A」だったらカウントアップ
 - indexOfで前から順に見て行って、見つかったらカウントアップし、その場所を起点に次の「A」を探す



ここから

- 課題提示
- 宿題の提示
- 課題の提出
- 課題確認できたら解散