

【解答】

【設問1】 aーア, bーユ  
【設問2】 cーウ, dーウ, eーカ, fーイ

【解説】

二つの文字列の差異を測る指標である編集距離をテーマに、エディットグラフの作成手順及びアルゴリズムについて考察する問題である。設問1は、二つの文字列間の編集距離を返す関数 CalceditDistance の実行結果に関する問題である。なじみのないテーマの問題ではあるが、エディットグラフの作成方法や編集距離の求め方については問題文に丁寧に説明されており、前提の知識がなくても解答は可能である。例年、問8のアルゴリズムの問題は、問題のテーマに関する知識がなくても、問題文から処理手順を理解すれば解答が可能な形式になっている。したがって、正解するためには、図も含め、書かれていることを読み解き、理解する力が求められる。また、プログラムは配列を用いた処理が多いため、問題文を理解する力を身に付けると同時に、配列の操作についてもしっかりと理解しておきたい。

はじめに、問題文の例を使って、どのように編集距離を求めているのかを確認する。文字列 "abcabba" を文字列 "cbabac" に変換するので、Str1[] と Str2[] は次のようになる。

Str1	0	1	2	3	4	5	6
	a	b	c	a	b	b	a

Str1Len=7

Str2	0	1	2	3	4	5
	c	b	a	b	a	c

Str2Len=6

編集距離は、エディットグラフと呼ばれるグラフの最短距離取得問題の考え方に基づいて求める。具体的な方法は説明の②に記述されている。最初に xy 平面上にエディットグラフを作成する。①の(a)に使い、点(X, 0)から点(X, Str2Len)に線分を引く。Xは  $0 \leq X \leq \text{Str1Len}$  を満たす全ての整数である。Str1Len は Str1[] の文字数なので7, Str2Len は Str2[] の文字数なので6ある。よって、線分は点(0, 0)から点(0, 6), 点(1, 0)から点(1, 6), ..., 点(6, 0)から点(6, 6), 点(7, 0)から点(7, 6)に引くことになる(図A)。続いて、(b)に従い、点(0, Y)から点(Str1Len, Y)に線分を引く。(a)と同様に、点(0, 0)から点(7, 0), ..., 点(0, 6)から点(7, 6)へと順に各点を結ぶ線を引く(図B)。

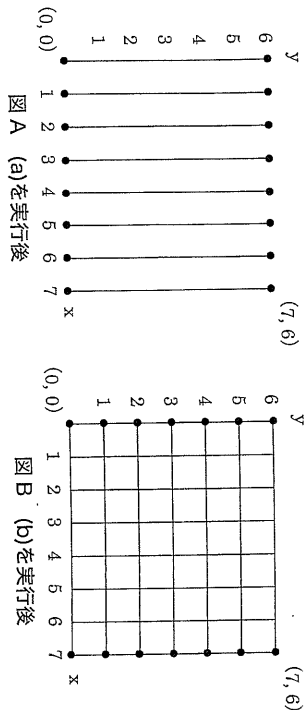


図 A (a)を実行後

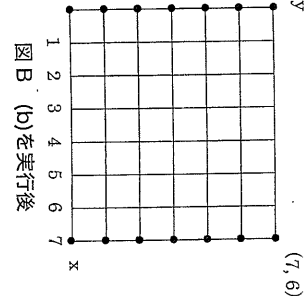


図 B (b)を実行後

①の(c)では、 $0 \leq X < \text{Str1Len}$ ,  $0 \leq Y < \text{Str2Len}$  を満たす全ての整数 X, Y の組に対して、Str1[X] と Str2[Y] が同一の文字の場合に、点(X, Y)から点(X + 1, Y + 1)に線分を引く。全ての X, Y の組なので、Str1[0] と Str2[0], Str1[0] と Str2[1], ..., Str1[6] と Str2[4], Str1[6] と Str2[5] を順に比較することになる。文字列 "abcabba" と文字列 "cbabac" では、Str1[0] と Str2[2] は同じ文字 'a' なので、点(0, 2)から点(1, 3)へ線分を引く。また、Str1[0] と Str2[4] も同じ文字なので、点(0, 4)から点(1, 5)にも線分を引く(図C)。このように全ての文字を比較して、文字が同じ場合に線分を引いていくと、図Dのようになる。

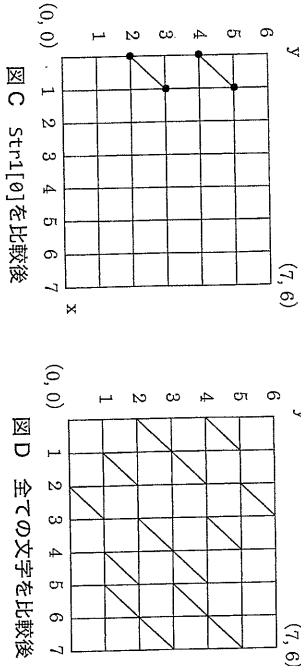


図 C Str1[0]を比較後

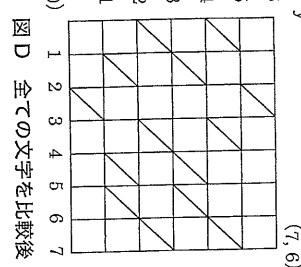


図 D 全ての文字を比較後

②は、編集距離を求める手順である。図Dのエディットグラフを構成する線分をたどって、点(0, 0)から点(Str1Len, Str2Len) = (7, 6)へ移動する経路を考える。このとき、点(X, Y)から点(X + 1, Y)又は点(X, Y + 1)への移動距離は1, 点(X, Y)から点(X + 1, Y + 1)への移動距離は0とする。X と Y をともに0のときを例として考えると、点(0, 0)から点(1, 0), 点(0, 0)から点(0, 1)の移動、つまり、横と縦の移動の場合は、距離は1, 点(0, 0)から点(1, 1)の移動、つまり右斜め上への移動の場合は、距離は0となる。このことから、右斜め上に移動する方が編集距離は短くなること分かる。このようにして最短移動距離を求めた結果、最短移動距離のーつは問題の図2右側のような経路となる。

次に、プログラムについて確認する。

【プログラム】

```
○整数型: CalceditDistance (文字型: Str1[], 整数型: Str1Len,
文字型: Str2[], 整数型: Str2Len)
○整数型: D[Str1Len + 1, Str2Len + 1], X, Y
```

```
01  X: 0, X ≤ Str1Len, 1
02  D[X, 0] ← X
03
04  Y: 0, Y ≤ Str2Len, 1
05  D[0, Y] ← Y
06
07  X: 1, X ≤ Str1Len, 1
08  Y: 1, Y ≤ Str2Len, 1
09  D[X, Y] ← Min (D[X - 1, Y - 1],
10                    D[X, Y - 1] + 1, D[X - 1, Y] + 1) ← α
11
12  D[X, Y] ← Min (D[X, Y - 1] + 1, D[X - 1, Y] + 1) ← β
13
14  return ( D[X, Y] )
15
```

/\* 編集距離を返却値として返す \*/

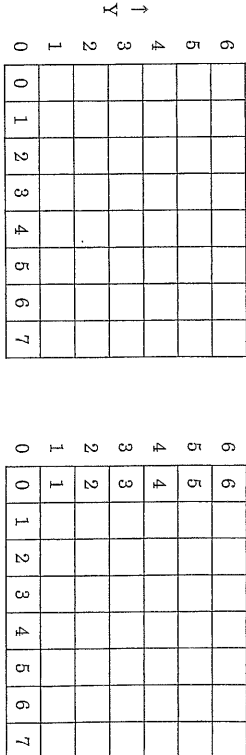


図 E 行番号 01~03 の繰返し処理を行

った後の配列 D の内容

※分かりやすいように、X と Y の向きをエディットグラフに合わせてある。

D[X, Y] には、点(0, 0)から点(X, Y)への最短移動距離が格納される。例えば、点(0, 0)から点(2, 2)への最短移動距離は、D[2, 2] に格納される。関数 CalceditDistance は、既に算出されている D[X - 1, Y - 1], D[X, Y - 1], D[X - 1, Y] を用いて D[X, Y] を求める関数である。問題文の例のエディットグラフと配列 D の対応は図 G のようになる。エディットグラフ上の点(X, Y)は、配列 D では D[X, Y] に対応する。また、X 方向が Str1[], Y 方向が Str2[] の文字列と対応し、配列の添字は 0 から始まることに注意する。

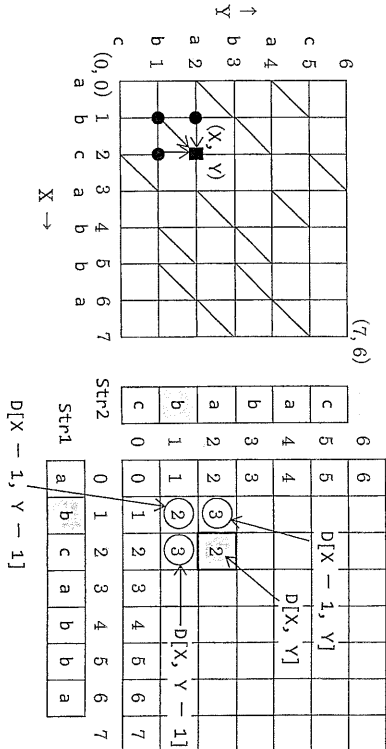


図 G エディットグラフと配列 D の対応

【設問1】

・空欄 a: 分岐のための条件式である。解答群を見ると、選択肢は全て Str1[] と Str2[] の文字の比較を行うものである。また、この空欄の条件による分岐後の処理を見ると、真の場合の処理 (行番号 10) と偽の場合の処理 (行番号 11) の違いは、関数 Min の引数の中に、D[X - 1, Y - 1] を含むか、含まないかである。仮に X=2, Y=2 として、行番号 10 の D[X - 1, Y - 1], D[X, Y - 1] + 1, D[X - 1, Y] + 1 の X と Y に 2 を代入すると、それぞれ、D[1, 1], D[2, 1] + 1, D[1, 2] + 1 となる。そして、図 G から分かるようにに、点(2, 2)と線分で結ぶことができる点は(1, 1), (2, 1), (1, 2)の3点である。ただし、点(1, 1)から点(2, 2)への線分が存在するのは、Str1[1] と Str2[1] の文字が同じ場合だけである。このときの更新対象である D[2, 2] と Str1[1], Str2[1] の添字を X と Y を使って表すと、それぞれ、D[X, Y], Str1[X - 1] と Str2[Y - 1] となる。したがって、関数 Min の引数の中に D[X - 1, Y - 1] を含むのは Str1[X - 1] = Str2[Y - 1] の場合であり、空欄 a の正解は (ア) になる。

なお、説明の②では、「点(X, Y)から点(X + 1, Y)又は点(X, Y + 1)への移動距離を1, 点(X, Y)から点(X + 1, Y + 1)への移動距離は0としたとき」というように、移動元を点(X, Y)としている。一方、プログラムでは移動先を点(X, Y)としているので、点(X - 1, Y)から点(X, Y), 点(X, Y - 1)から点(X, Y), 点(X - 1, Y - 1)から点(X, Y)への移動と置き換えて考えるとよい。

・空欄 b: 関数 CalceditDistance の返却値は、二つの文字列間の編集距離である。説明の②に「点(0, 0)から点(Str1Len, Str2Len)までの最短移動距離が編集距離となる」と記述されているので、その値が入っている (ウ) が正解である。

【設問2】

・空欄 c: Str1[] = "peace", Str2[] = "people"なので、Str1Len=5, Str2Len=6 となる。先に説明したようにに、Str1[0] と Str2[0] から順に全ての文字の比較を行い、文字が同じ場合に線分を引く。最初は、Str1[0] と Str2[0] が同じ文字なので、点(0, 0)から点(1, 1)に線分を引くことになる。続けて文字を順に比較していくと、次の表のような結果になる。したがって、(ウ) が正解である。

【解答】

【設問 1】 aーア, bーウ  
【設問 2】 cーウ, dーウ, eーカ, fーイ

【解説】

二つの文字列の差異を測る指標である編集距離をテーマに、エディットグラフの作成手順及びアルゴリズムについて考察する問題である。設問 1 は、二つの文字列間の編集距離を返す関数 CalcEditDistance の実行結果に関する問題である。なじみのないテーマの問題ではあるが、エディットグラフの作成方法や編集距離の求め方については問題文に丁寧に説明されており、前提の知識がなくても解答は可能である。例年、問 8 のアルゴリズムの問題は、問題のテーマに関する知識がなくても、問題文から処理手順を理解すれば解答が可能な形式になっている。したがって、正解するためには、図も含め、書かれていることを読み解き、理解する力が求められる。また、プログラムは配列を用いた処理が多いため、問題文を理解する力を身に付けると同時に、配列の操作についてもしっかりと理解しておきたい。

はじめに、問題文の例を使って、どのように編集距離を求めているのかを確認する。文字列 "abcabba" を文字列 "cbabac" に変換するので、Str1[] と Str2[] は次のようになる。

Str1	0	1	2	3	4	5	6
	a	b	c	a	b	b	a

Str1Len=7

Str2	0	1	2	3	4	5
	c	b	a	b	a	c

Str2Len=6

編集距離は、エディットグラフと呼ばれるグラフの最短距離取得問題の考え方に基づいて求める。具体的な方法は説明の(2)に記述されている。最初に xy 平面上にエディットグラフを作成する。①の(a)に使い、点(X, 0)から点(X, Str2Len)に線分を引く。X は  $0 \leq X \leq \text{Str1Len}$  を満たす全ての整数である。Str1Len は Str1[] の文字数なので 7、Str2Len は Str2[] の文字数なので 6 ある。よって、線分は点(0, 0)から点(0, 6)、点(1, 0)から点(1, 6)、…、点(6, 0)から点(6, 6)、点(7, 0)から点(7, 6)に引くことになる(図 A)。続いて、(b)に従い、点(0, Y)から点(Str1Len, Y)に線分を引く。(a)と同様に、点(0, 0)から点(7, 0)、…、点(0, 6)から点(7, 6)へと順に各点を結ぶ線を引く(図 B)。

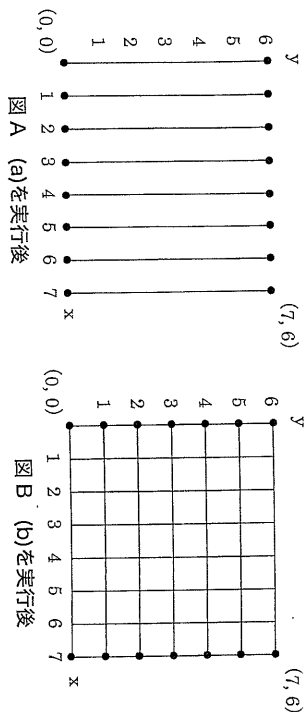


図 B (b)を実行後

①の(c)では、 $0 \leq X < \text{Str1Len}$ ,  $0 \leq Y < \text{Str2Len}$  を満たす全ての整数 X, Y の組に対して、Str1[X] と Str2[Y] が同一の文字の場合に、点(X, Y)から点(X + 1, Y + 1)に線分を引く。全ての X, Y の組なので、Str1[0] と Str2[0] と Str1[1], ..., Str1[6] と Str2[4], Str1[6] と Str2[5] を順に比較することになる。文字列 "abcabba" と文字列 "cbabac" では、Str1[0] と Str2[2] は同じ文字 'a' なので、点(0, 2)から点(1, 3)へ線分を引く。また、Str1[0] と Str2[4] も同じ文字なので、点(0, 4)から点(1, 5)にも線分を引く(図 C)。このように全ての文字を比較して、文字が同じ場合に線分を引いていくと、図 D のようになる。

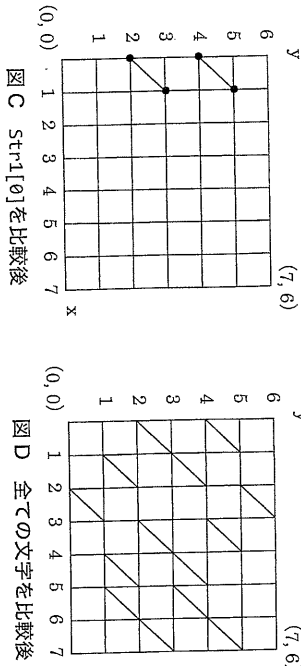


図 D 全ての文字を比較後

②は、編集距離を求める手順である。図 D のエディットグラフを構成する線分をたどって、点(0, 0)から点(Str1Len, Str2Len) = (7, 6)へ移動する経路を考える。このとき、点(X, Y)から点(X + 1, Y)又は点(X, Y + 1)への移動距離は 1、点(X, Y)から点(X + 1, Y + 1)への移動距離は 0 とする。X と Y をともに 0 のときを例として考えると、点(0, 0)から点(1, 0)、点(0, 0)から点(0, 1)の移動、つまり、横と縦の移動の場合は、距離は 1、点(0, 0)から点(1, 1)の移動、つまり右斜め上への移動の場合は、距離は 0 となる。このことから、右斜め上に移動する方が編集距離は短くなること分かる。このようにして最短移動距離を求めた結果、最短移動距離のーつは問題の図 2 右側のような経路となる。

次に、プログラムについて確認する。

【プログラム】

```
○整数型: CalcEditDistance (文字型: Str1[], 整数型: Str1Len,
文字型: Str2[], 整数型: Str2Len)
○整数型: D[Str1Len + 1, Str2Len + 1], X, Y
```

```
01  X: 0, X ≤ Str1Len, 1
02  D[X, 0] ← X
03
04  Y: 0, Y ≤ Str2Len, 1
05  D[0, Y] ← Y
06
07  X: 1, X ≤ Str1Len, 1
08  Y: 1, Y ≤ Str2Len, 1
09  D[X, Y] ← Min (D[X - 1, Y - 1],
10                    D[X, Y - 1] + 1, D[X - 1, Y] + 1) ← α
11
12  D[X, Y] ← Min (D[X, Y - 1] + 1, D[X - 1, Y] + 1) ← β
13
14  return ( D[X, Y] )
15
```

/\* 編集距離を返却値として返す \*/

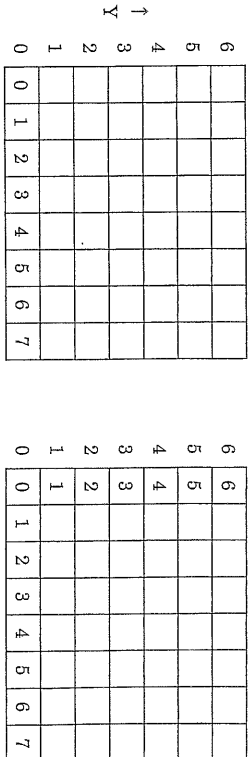


図 F 行番号 04~06 の繰返し処理を行った後の配列 D の内容

※分かりやすいように、X と Y の向きをエディットグラフに合わせてある。

D[X, Y] には、点(0, 0)から点(X, Y)への最短移動距離が格納される。例えば、点(0, 0)から点(2, 2)への最短移動距離は、D[2, 2] に格納される。関数 CalcEditDistance は、既に算出されている D[X - 1, Y - 1], D[X, Y - 1], D[X - 1, Y] を用いて D[X, Y] を求める関数である。問題文の例のエディットグラフと配列 D の対応は図 G のようになる。エディットグラフ上の点(X, Y)は、配列 D では D[X, Y] に対応する。また、X 方向が Str1[], Y 方向が Str2[] の文字列と対応し、配列の添字は 0 から始まることに注意する。

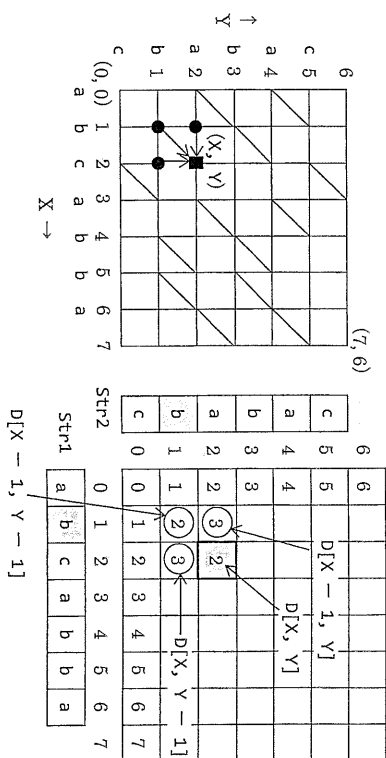


図 G エディットグラフと配列 D の対応

【設問 1】

・空欄 a: 分岐のための条件式である。解答群を見ると、選択肢は全て Str1[] と Str2[] の文字の比較を行うものである。また、この空欄の条件による分岐後の処理を見ると、真の場合の処理 (行番号 10) と偽の場合の処理 (行番号 11) の違いは、関数 Min の引数の中に、D[X - 1, Y - 1] を含むか、含まないかである。仮に X=2, Y=2 として、行番号 10 の D[X - 1, Y - 1], D[X, Y - 1] + 1, D[X - 1, Y] + 1 の X と Y に 2 を代入すると、それぞれ、D[1, 1], D[2, 1] + 1, D[1, 2] + 1 となる。そして、図 G から分かるように、点(2, 2)と線分で結ぶことができる点は(1, 1), (2, 1), (1, 2)の 3 点である。ただし、点(1, 1)から点(2, 2)への線分が存在するのは、Str1[1] と Str2[1] の文字が同じ場合だけである。このときの更新対象である D[2, 2] と Str1[1], Str2[1] の添字を X と Y を使って表すと、それぞれ、D[X, Y], Str1[X - 1] と Str2[Y - 1] となる。したがって、関数 Min の引数の中に D[X - 1, Y - 1] を含むのは Str1[X - 1] = Str2[Y - 1] の場合であり、空欄 a の正解は (ア) になる。

なお、説明の②では、「点(X, Y)から点(X + 1, Y)又は点(X, Y + 1)への移動距離を 1、点(X, Y)から点(X + 1, Y + 1)への移動距離は 0 としたとき」というように、移動元を点(X, Y)としている。一方、プログラムでは移動先を点(X, Y)としているので、点(X - 1, Y)から点(X, Y)、点(X, Y - 1)から点(X, Y)、点(X - 1, Y - 1)から点(X, Y)への移動と置き換えて考えるとよい。

・空欄 b: 関数 CalcEditDistance の返却値は、二つの文字列間の編集距離である。説明の②に「点(0, 0)から点(Str1Len, Str2Len)までの最短移動距離が編集距離となる」と記述されているので、その値が入っている (ウ) が正解である。

【設問 2】

・空欄 c: Str1[] = "peace", Str2[] = "people" なので、Str1Len=5, Str2Len=6 となる。先に説明したように、Str1[0] と Str2[0] から順に全ての文字の比較を行い、文字が同じ場合に線分を引く。最初は、Str1[0] と Str2[0] が同じ文字なので、点(0, 0)から点(1, 1)に線分を引くことになる。続けて文字を順に比較していくと、次の表のような結果になる。したがって、(ウ) が正解である。