

# ABC 380 A - 123233

hiragn

2024 年 12 月 12 日

## 1. 問題の概要

6 桁の自然数  $N$  が与えられる。 $N$  の各位の数字の 1 つが 1, 2 つが 2, 3 つが 3 であるかどうか判定せよ。

[https://atcoder.jp/contests/abc380/tasks/abc380\\_a](https://atcoder.jp/contests/abc380/tasks/abc380_a)

「ソートして 122333 と比較」「Count で数える」「Counts で連想を作って数える」の 3 通りの方法で解いた。

Count と Counts は紛らわしい。python でいえば前者は count メソッドで、後者は dict (連想配列) にあたる。連想にデフォルト値を設定する方法がわからなくて苦労したが、Lookup で処理できた。

## 2. ソートして 122333 と比較

多分一番単純な方法はこれ。

1.  $N$  を桁数字のリストに直してソート
2. FromDigits で連結して 122333 と比較

連結するかわりに 122333 を桁数字のリストに変換してもいいが、このやり方のほうが素直だと思う。

---

```
1 In[] := Clear["Global`*"];
2 f[n_] := FromDigits@Sort@IntegerDigits@n == 122333;
3 f /@ {123233, 123234, 323132, 500000}
4
5 Out[] = {True, False, True, False}
```

---

### 3. Count で数える

Count を使って 1, 2, 3 の個数を数える。AllTrue を使うと、この処理はまとめて書ける。

---

```

1 In[] := Clear["Global`*"];
2 f[n_] := AllTrue[Range@3, Count[IntegerDigits@n, #] == # &];
3 f /@ {123233, 123234, 323132, 500000}
4
5 Out[] = {True, False, True, False}

```

---

### 4. Counts で連想を作って数える

桁数字のリストに Counts を使うと「key = 数字, value = その個数」の辞書 (mathematica では「連想」という) ができる。

存在しない key を指定すると Missing["KeyAbsent", key] が返ってくるのが使いづらい。

---

```

1 In[] := n = 122333;
2 asc = Counts@IntegerDigits@n
3 Out[] = <|1 -> 1, 2 -> 2, 3 -> 3|>
4
5 In[] := asc /@ Range@3
6 Out[] = {1, 2, 3}
7
8 In[] := asc@4
9 Out[] = Missing["KeyAbsent", 4]

```

---

python の defaultdict のようにデフォルト値を指定する方法を調べて、Lookup を使えばいいことがわかった。この関数は 3 つの引数を取り、3 つ目の引数がデフォルト値になる。

たとえば 5000 の桁数字のリストに対して 0~9 の個数を調べると次のようになる。

---

```

1 In[] := n = 5000;
2 Lookup[Counts@IntegerDigits@n, Range[0, 9], 0]
3
4 Out[] = {3, 0, 0, 0, 0, 0, 1, 0, 0, 0}

```

---

これと AssociationThread を使って新たな連想を作ると defaultdict 的なものができる。

---

```
1 In[]:= Clear["Global`*"];
2 f[n_] := Module[{asc},
3   asc =
4     AssociationThread[
5       Range[0, 9] -> Lookup[Counts@IntegerDigits@n, Range[0, 9], 0]];
6   AllTrue[Range@3, asc@# == # &]];
7 f /@ {123233, 123234, 323132, 500000}
8
9 Out[] = {True, False, True, False}
```

---