2015 年度 大問 5

hari64boli64 (hari64boli64@gmail.com)

2025 年 4 月 23 日

1 問題

$$D_i = \{j \in \mathbb{N} | 1 \leq j < i, S[j] = S[i]\}$$

$$d(i) = \begin{cases} i - \max D_i & (D_i \neq \emptyset) \\ i & (D_i = \emptyset) \end{cases}$$

2 解答

5.py が解答。

Listing 1 answer

```
import math
import random
from collections import defaultdict

random.seed(0)


def makeProblem(size):
    sigma = random.randint(1, size)
    n = random.randint(1, size)
    A = list(range(1, sigma + 1))
    S = [random.choice(A) for _ in range(n)]
    k = len(set(S))
    return sigma, n, S, k


def problem1(sigma, n, S):
```

```
18      # 問題(1)の解答
19
20      # 1. max(D_i)を求めるのに必要な、
21      # 各の要素に関して最後の出現位置を記録する配列を用意する    A
22      # 空間    O(sigma) 時間 O(sigma)
23      last = [0 for _ in range(sigma)]
24
25      # 2. 各について、id(i)を計算する
26      # 空間    O(1) 時間 O(n)
27      for i in range(n):
28          yield (i + 1) - last[S[i] - 1]
29          last[S[i] - 1] = i + 1
30
31
32  def problem2(sigma, n, S):
33      # 問題(2)の解答
34
35      # 1. max(D_i)を求めるのに必要な、
36      # 各の要素に関して最後の出現位置を記録する配列を辞書で持つ    A
37      # 空間    O(k) 時間 O(1)
38      last = defaultdict(int)
39
40      # 2. 各について、id(i)を計算する
41      # 空間    O(1) 時間 O(n logk)
42      for i in range(n):
43          # ここの計算量が O(logk) になる
44          yield (i + 1) - last[S[i] - 1]
45          last[S[i] - 1] = i + 1
46
47
48  def problem3(sigma, n, S):
49      # 問題(3)の解答
50
51      # ならし計算量の解析に近い
52      # に現れる要素それぞれに注目して見ると、S
53      # 総計で見る回数は必ずで抑えられるn
54      # よって、全体でO(nk)
55
56      for i in range(n):
57          for j in range(i - 1, -1, -1):
58              if S[j] == S[i]:
59                  yield i - j
60                  break
61          else:
62              yield i + 1
63
64
```

```python
def trial():
    sigma, n, S, k = makeProblem(size=10)
    print(f"{sigma=}, {n=}, {k=}")
    print(f"{S=}")

    Ds = [[j + 1 for j in range(i) if S[j] == S[i]] for i in range(n)]
    ds = [i - max(D) if D else i for i, D in enumerate(Ds, start=1)]

    print(f"{Ds=}")
    print(f"{ds=}")

    ans1 = list(problem1(sigma, n, S))
    ans2 = list(problem2(sigma, n, S))
    ans3 = list(problem3(sigma, n, S))

    assert ds == ans1, ans1
    assert ds == ans2, ans2
    assert ds == ans3, ans3


def computeSumOfd():
    # 問題(4)の解答

    #   \sum d <= kn
    # ⇒ 1/n \su d <= k
    # の凸性より、log
    #   1/n \sum log d <= log (1/n \sum d) <= log k
    # ⇒ \sum log d <= n log k


    sigma, n, S, k = makeProblem(size=10000)
    print(f"{sigma=}, {n=}, {k=}")

    ds = list(problem1(sigma, n, S))

    print(f"{sum(math.log(d) for d in ds)=}")
    print(f"{n*math.log(k)=}")


def main():
    for i in range(5):
        trial()
        if i != 4:
            print("-" * 10)

```

```
    print("=" * 10)

    for i in range(5):
        computeSumOfd()
        if i != 4:
            print("-" * 10)


if __name__ == "__main__":
    main()
```

Listing 2 output

```
sigma=7, n=7, k=5
S=[1, 3, 5, 4, 4, 7, 7]
Ds=[[], [], [], [], [4], [], [6]]
ds=[1, 2, 3, 4, 1, 6, 1]
----------
sigma=5, n=8, k=4
S=[3, 5, 2, 5, 2, 3, 2, 1]
Ds=[[], [], [], [2], [3], [1], [3, 5], []]
ds=[1, 2, 3, 2, 2, 5, 2, 8]
----------
sigma=10, n=5, k=5
S=[9, 10, 3, 5, 2]
Ds=[[], [], [], [], []]
ds=[1, 2, 3, 4, 5]
----------
sigma=2, n=6, k=2
S=[2, 1, 2, 2, 2, 1]
Ds=[[], [], [1], [1, 3], [1, 3, 4], [2]]
ds=[1, 2, 2, 1, 1, 4]
----------
sigma=9, n=8, k=6
S=[8, 9, 5, 1, 9, 1, 2, 7]
Ds=[[], [], [], [], [2], [4], [], []]
ds=[1, 2, 3, 4, 3, 2, 7, 8]
==========
sigma=19, n=8087, k=19
sum(math.log(d) for d in ds)=19935.714435979888
n*math.log(k)=23811.678024519002
----------
sigma=1397, n=38, k=35
sum(math.log(d) for d in ds)=96.64773012539216
n*math.log(k)=135.10322633659771
----------
sigma=8410, n=3796, k=3077
sum(math.log(d) for d in ds)=26698.671661842043
```

```
n*math.log(k)=30488.372584722474
----------
sigma=7194, n=6237, k=4169
sum(math.log(d) for d in ds)=45840.86686343042
n*math.log(k)=51988.08612754253
----------
sigma=8530, n=4156, k=3267
sum(math.log(d) for d in ds)=29505.23672883638
n*math.log(k)=33628.80352261405
```