

2016 年度 大問 5

hari64boli64 (hari64boli64@gmail.com)

2025 年 4 月 23 日

1 問題

累積積、SegmentTree

2 解答

[5.py](#) が解答。

Listing 1 answer

```
1 import math
2 import random
3 import numpy as np
4 from typing import List
5
6
7 def makeProblem(problemNo: int):
8     assert 1 <= problemNo <= 4
9     zeroRate = random.random()
10    generator = random.random
11    n = random.randint(1, 10)
12    if problemNo == 1:
13        zeroRate = 0.0
14    if problemNo == 2:
15        generator = lambda: random.randint(0, 1)
16    if problemNo == 4:
17        n = 16
18        zeroRate = 1 / n
19        generator = lambda: random.randint(1, 5)
20    As = [0 if random.random() < zeroRate else generator() for _
21          in range(n)]
22    return n, As
23
```

```

24 def check(n: int, As: List[int], solver: callable):
25     # がアルゴリズムf, がデータ構造に対応するdata
26     f, data = solver(n, As)
27     # 全てのクエリに対して正しい答えが返ってくるかチェックする
28     for s in range(1, n + 1):
29         for t in range(s, n + 1):
30             ans = f(s, t, data)
31             print(s, t, ans)
32             if not np.isclose(ans, math.prod(As[s - 1 : t])):
33                 raise Exception("Wrong Answer")
34     print("ok!")
35
36
37 def solve1(n: int, As: List[int]) -> callable:
38     """
39     \prod_{i=s}^t A[i] =
40     (\prod_{i=1}^t A[i]) / (\prod_{i=1}^{s-1} A[i])
41     """
42     Ps = [1.0]
43     for i in range(n):
44         Ps.append(Ps[-1] * As[i])
45
46     def f(s: int, t: int, Ps: List[int]):
47         return Ps[t] / Ps[s - 1]
48
49     return f, Ps
50
51
52 def solve2(n: int, As: List[int]) -> callable:
53     """の個数を数えて、
54     0
55     [s,t]間でその個数の増減がないかを調べれば良い
56     """
57     numOfZeros = [0]
58     for i in range(n):
59         numOfZeros.append(numOfZeros[-1] + int(As[i] == 0))
60
61     def f(s: int, t: int, numOfZeros: List[int]):
62         return 0 if numOfZeros[t] != numOfZeros[s - 1] else 1
63
64     return f, numOfZeros
65
66
67 def solve3(n: int, As: List[int]) -> callable:
68     """
69     (1)と(2)を組み合わせるだけで良いは、が出てきたらリセットの意味でにする
70     Ps01

```

```

71     """
72     Ps = [1.0]
73     for i in range(n):
74         Ps.append(Ps[-1] * As[i])
75         if Ps[-1] == 0:
76             Ps[-1] = 1
77
78     numOfZeros = [0]
79     for i in range(n):
80         numOfZeros.append(numOfZeros[-1] + int(As[i] == 0))
81
82     def f(s: int, t: int, Ps_numOfZeros: List[int]):
83         Ps, numOfZeros = Ps_numOfZeros
84         return 0 if numOfZeros[t] != numOfZeros[s - 1] else Ps[t]
85             / Ps[s - 1]
86
87     return f, (Ps, numOfZeros)
88
89 def vis4():
90     """、もしくは、で検索
91     SegmentTreeBinaryIndexedTree空間計算量は、
92      $O(\log N)$ のように思えるが、よくよく考えると、
93      $2N$ で抑えられている-1出力例
94
95     :
96
97     As=[3, 1, 4, 1, 3, 3, 3, 2, 1, 3, 3, 0, 5, 4, 3, 5] 段目
98
99     0: 0-----段目
100    1: 648----- 0-----段目
101    2: 12----- 54----- 0-----300-----段目
102    3: 3--- 4--- 9--- 6--- 3--- 0--- 20--- 15---段目
103    4: 3 1 4 1 3 3 3 2 1 3 3 0 5 4 3 5
104    """
105    n, As = makeProblem(4)
106    print(f"{As=}")
107    logN = round(math.log2(n))
108    for i in range(logN + 1):
109        print(f"{i}段目: ", end="")
110        for j in range(2**i):
111            print(
112                f"{math.prod(As[(2**(logN-i)) * j : (2**(logN-i)) * (j+1)]) : >3}",
113                end=" " * (3 * ((2 ** (logN - i) - 1))),
114            )
115    print()

```

```

116
117
118 def main():
119     for _ in range(10):
120         n, As = makeProblem(1)
121         print(f"{n=}, {As=}")
122         check(n, As, solve1)
123     for _ in range(10):
124         n, As = makeProblem(2)
125         print(f"{n=}, {As=}")
126         check(n, As, solve2)
127     for _ in range(10):
128         n, As = makeProblem(3)
129         print(f"{n=}, {As=}")
130         check(n, As, solve3)
131     vis4()
132
133
134 if __name__ == "__main__":
135     main()

```