РЕШЕНИЕ НЕКОТОРЫХ АЛГОРИТМИЧЕСКИХ ЗАДАЧ НА САЙТЕ CODEFORCES

Рустамова Нигина Бобир кизи *- Студентка,*
Бухарский Государственный университет,
Узбекистан, г. Бухара
Рустамов Хаким Шарипович *- доцент,*
Бухарский государственный университет,
Узбекистан, г. Бухара

АННОТАЦИЯ

В этой статье рассматриваются некоторые задачи и их решения с использованием языка программирования Python. Такие задания повышают знания юниора и помогают ему глубоко мыслить. В статье обсуждается трудоемкий процесс решения задач и это помогает увеличить опыт разработчика. Все задачи, описанные в этой статье, взяты из сайта Codeforces, мы решили эти задачи, теперь делимся с вами решением этой задачи. Здесь мы рассмотрим 4 задачи и их решения. Решая подобные задачи, вы сможете найти более простые пути, и это очень поможет вам расширить свои знания.

Ключевые слова: mn- имя переменной, содержащее минимальное значение списка, y- имя переменной, счетчик

Codeforces — проект, объединяющий людей, которые интересуются и участвуют в соревнованиях по программированию. С одной стороны Codeforces является социальной сетью, посвященной программированию и соревнованиям по программированию. С другой стороны — это площадка, где регулярно проводятся соревнования, навыки участников отражает рейтинг, а прошедшие соревнования могут быть использованы для подготовки. Codeforces постоянно развивается, и в наших планах усовершенствование платформы для предоставления возможностей проводить контесты участникам самостоятельно, наполнение проекта учебным контентом, развитие Codeforces как тренировочной и учебной платформы.[1]

Мы рассмотрим задачи различной сложности. Решение алгоритмических задач — отличный способ улучшить свой опыт по программирования. Задачи из Codeforces Round 898 (Div. 4). Рассмотрим первую задачу под названием A.«Короткая сортировка».

Есть три карты с буквами a, b, c, расположенные в ряд в некотором порядке. Вы можете выполнить следующую операцию не более одного раза:

Выберите две карты и поменяйте их местами.

Возможно ли, чтобы ряд стал abc после выполнения операции? Выведите «YES», если это возможно, и «NO» в противном случае.

Примечание

В первом наборе входных данных примера нам не нужно выполнять никаких операций, так как ряд уже abc.

Во втором наборе входных данных примера мы можем поменять местами c и b: acb→abc.

В третьем наборе входных данных примера мы можем поменять местами b и a: bac→abc.

В четвертом наборе входных данных примера невозможно получить abc с помощью не более одной операции.

Условие гласит, что нам нужно найти комбинации, которые можно преобразовать в вид abc всего за одну операцию. Таких комбинаций очень мало и поэтому все эти комбинации мы можем записать и проверить с помощью условного оператора. Мы можем разработать алгоритм для этой задачи, но это очень хорошее решение, к тому же этот метод выполняется за меньшее время и требует меньше памяти. Перейдем к написанию кода.

A. Короткая сортировка

```
a=int(input())
s=''
for i in range(a):
s=input()
if s=='abc' or s=='acb' or s=='bac' or s=='cba':
print('YES')
else:
print('NO')
```

Теперь перейдем к решению нашей второй задачи (B. Хороший мальчик).

Славик готовит подарок для дня рождения друга. У него есть массив a из n цифр, и подарком будет произведение всех этих цифр. Поскольку Славик - хороший ребенок, он хочет сделать наибольшее возможное произведение, для этого он может добавить 1 к ровно одной из своих цифр.

Какое максимальное произведение может получить Славик?

Входные данные

Первая строка содержит одно целое число t ($1 \le t \le 10^4$) — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число n ($1 \le n \le 9$) — количество цифр.

Вторая строка каждого набора содержит n целых чисел, разделенных пробелом, a_i ($0 \le a_i \le 9$) — цифры в массиве.

Выходные данные

Для каждого набора входных данных выведите одно целое число — максимальное произведение, которое может получить Славик, добавив 1 к ровно одной из своих цифр.

Здесь все очень доходчиво объяснено, нет неясных моментов. Только нужно думать, на какое из этих чисел можно сложить единицу чтобы умножение получилось максимально. Попробуем добавить эту единицу на самую максимальному числу и вычислить умножение. Тесты для нашей задачи.

4
4
2 2 1 2
3
0 1 2
5
4 3 2 3 4
9
9 9 9 9 9 9 9 9 9
Ответы:
16
2
432
430467210

В первом случае максимальное значение — 2, но количества двоек тут три , поэтому прибавляем эту единицу к одну из двойки (2+1)*2*1*2=12.

Теперь второй пример: 0*1*(2+1)=0.

Третий пример: (4+1)*3*2*3*4=360.

Четвёртый пример: (9+1)*9*9*9*9*9*9*9*9=430 467 210.

Только один из наших ответов подходят а другие не соответствует ответам в примере, поэтому нам придется думать по другому. Попробуем

прибавить эту единицу к самому наименьшему числу. В этом и есть идея нашего второго примера, ведь самое маленькое число там — это ноль, а когда мы прибавляем единицу, этот ноль становится 1, и в результате умножения мы получаем большее число. Теперь будем проверить эту идею на других примерах.

Первый пример: 2*2*(1+1)*2=16 Правильно.

Второй пример: (0+1)*1*2=2

Третий пример: 4*3*(2+1)*3*4=432

Четвёртый пример: (9+1)*9*9*9*9*9*9*9*9*9=430 467 210.

Идея у нас есть, теперь можно переходить к коду.

B. Хороший ребенок

```
a=int(input())
for i in range(a):
    b=int(input())
    l=list(map(int,input().split()))
    mn=min(l)
    l.remove(mn)
    s=1
    for j in l:
        s*=j
    print(s*(mn+1))
    s=1
```

Это код нашей задачи, где я сначала получила данные, затем удалила самое маленькое число и сохранила на переменную по имени mn, а затем нашла произведение оставшихся чисел. В конце я прибавила единицу к наименьшему числу и умножила на произведение остальных чисел.

Наша третья задача называется (C. Стрельба по мишени).

Мишень размером 10×10 состоит из пяти «колец», как показано на рисунке. Попадание в каждое кольцо даёт разное количество очков: внешнее кольцо — 1 очко, следующее кольцо — 2 очка, ..., центральное кольцо — 5 очков.

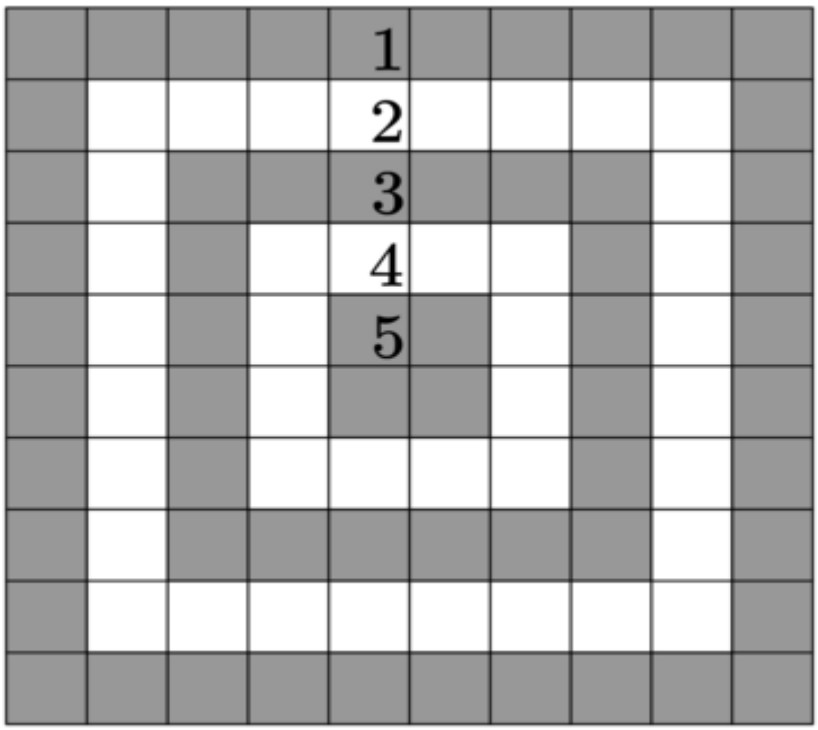

Рис.1

Влад выпустил несколько стрел по мишени. Помогите ему определить, сколько очков он набрал.

Входные данные

Ввод содержит несколько наборов входных данных. Первая строка ввода содержит одно целое число t (1≤t≤1000) — количество наборов входных данных.

Каждый набор состоит из 10 строк, каждая из которых содержит 10 символов. Каждый символ в сетке может быть либо X (обозначает попадание стрелы), либо (обозначает промах).

Выходные данные

Для каждого набора входных данных выведите одно целое число — общее количество очков.

У нас есть мишень и Влад сделал несколько выстрелов, мы считаем и показываем очки Влада. Каждая строка мишени имеет разные точки, например, если стрела попадает в большой квадрат мишени, Блад получает 1 очко, если попадает квадрат меньшего размера получает 2 очка и так далее. Для каждой строки нам нужно написать отдельные условия и добавить отдельные очки. Это повторяется 10 раз, потому что наш мишень имеет размер 10x10. Вы можете найти простой способ, но мой метод заключается в том, что я проверяю каждую

строку на определенное условие и добавляю определенные числа. Например, при проверке строки 5 или 6, если пуля попадает в первую ячейку, добавляется 1 очков, если она попадает во вторую ячейку, добавляются 2 очко и так далее. Если пуля попадает в 5 или 6 ячейку, я добавляю 5 очков, потому что наша самая большая очко это 5.

А что насчет остальных ячеек? Если пуля попадает в 7-ю ячейку, то прибавляется 4 очка, если пуля попадает в 8-ю ячейку, то прибавляется 3 очка. Если в ячейке в девятую ячейку, то прибавляется 2 очка. Если пуля попадает в 10-ю ячейку, добавляется 1 очко. Теперь перейдем к решению.

C. Стрельба по мишени

```
a=int(input())
s=0
x=0
for i in range(a):
    for j in range(1,11):
        b=input()
        for u in b:
            s+=1
            if u=='X':
                if j==1 or j==10:
                    x+=1
                elif j==2 or j==9:
                    if s==1 or s==10:
                        x+=1
                    else:
                        x+=2
                elif j==3 or j==8:
                    if s==1 or s==10:
                        x+=1
                    elif s==2 or s==9:
                        x+=2
                    else:
                        x+=3
                elif j==4 or j==7:
                    if s==1 or s==10:
                        x+=1
```

```
            elif s==2 or s==9:
                x+=2
            elif s==3 or s==8:
                x+=3
            else:
                x+=4
        elif j==5 or j==6:
            if s==1 or s==10:
                x+=1
            elif s==2 or s==9:
                x+=2
            elif s==3 or s==8:
                x+=3
            elif s==4 or s==7:
                x+=4
            else:
                x+=5
    s=0
print(x)
x=0
```

Этот код проверяет номер строки и номер ячейки и добавляет соответствующее значение в счетчик. Следующее задание это D. 1D Ластик.

У вас есть полоска бумаги s, которая имеет длину n ячеек. Каждая ячейка может быть либо черной, либо белой. За одну операцию вы можете выбрать любые k последовательных ячеек и сделать их все белыми.

Найдите минимальное количество операций, необходимых для удаления всех черных ячеек.

Примечание

В первом наборе входных данных примера вы можете выполнить следующие операции: WBWWWB→WWWWWB→WWWWWW

Во втором наборе входных данных примера вы можете выполнить следующие операции: WWBWBWW→WWWWWWW

В третьем наборе входных данных примера вы можете выполнить следующие операции: BWBWB→BWWWW→WWWWW

Очень сложный структурный пример, но вы можете решить эту задачу, подумав немного глубже. Обратите внимание на этой части задачи. В этой

задаче нам не нужно заменять все B на W, нам просто нужно найти минимальное количество операций, необходимое для удаления всех черных ячеек. Это значит, что мы проверим каждый элемент этой строки. Если мы находим буквы B, мы не проверяем следующие k элементов, но если наш элемент W, мы продолжаем проверку. Сначала я написала код, преобразующий строку в список, затем нахожу первую букву B, затем нахожу ее индекс и сохраняю в переменной ind. А потом от ind до ind+k я вырезаю из стартовой строки подстроку и добавляю единицу к счетчику, потом проверяю все элементы этой подстроки, если это элемент B, то меняю на W. Если вы пишете хороший код, этот алгоритм работает, но этот алгоритм требует много времени. Чтобы наш алгоритм работал быстро, нам не нужно найти индекс B или поменять B на W из подстроки ; если он найдет B, не проверяйте следующие k элементов и все. Теперь напишем код.

D. 1D Ластик

```
a=int(input())
for i in range(a):
    n,k=map(int,input().split())
    b=input()
    y=0
    s=0
    while s+1<=n:
        if b[s]=='B':
            y+=1
            s+=k
        else:
            s+=1
    print(y)
```

Здесь я проверяю, что переменная s не превысила длину строки. Затем я проверяю каждый элемент этого строки. Если этот элемент — B, я добавляю k к переменной s и добавляю единицу к счетчику y. Если этот элемент равен W, я добавляю 1 к переменной s.

Вывод

В этой статье мы рассмотрели решение различных задач на сайте Codeforces. Используя Python, мы преобразовали наши идеи в код. Чтобы улучшить свои знания в программировании, вам нужно решать разные задачи разными способами и это вам очень поможет. В этой статье мы обсудили

различные задачи, в некоторых задачах можно найти более простые пути и создать более быстрый алгоритм, и это здорово.

Список литературы:

1. codeforces.com