

Разбор задач

7-й ОМОШП “Пушинская новогодняя олимпиада”

Земли Острова Рапторов

23 декабря 2018 г.

СТАРТОВАЯ ЛИГА

Задача А. Высадка на остров

Автор задачи - Арсений Буланов

Задача предполагает умение участника работать со вводом/выводом и условным оператором.

Заметим, что если ветра нет ($W = 0$), то Паше всегда выгодно лететь на восток. Это выполняется, поскольку на востоке находится наиболее желаемое для него место, и, так как центральные джунгли имеют форму круга, то он либо может долететь до всех четырёх зон, помимо джунглей, либо не может долететь ни до одной из них (в силу симметричности задачи). Когда ветра нет ($W = 0$), возможны две ситуации:

1. Паша успевает долететь до границы ($H \geq 2R$). Паша падает на 2 метра каждую секунду, и пролетает в сторону 1 метр за ту же секунду. Таким образом, чтобы пролететь в сторону на длину R , ему нужно находиться на высоте хотя бы $2R$. Ответ - “forest”.
2. Обратная ситуация. Ответ - “jungle”.

Когда ветер есть, Паше имеет смысл лететь только по направлению ветра (если он будет лететь против ветра, то скорости ветра и Паши скомпенсируются и итоговая скорость будет равняться нулю, из-за чего тот упадет в джунгли, что является наихудшим исходом. Поскольку зоны к северу и к югу от Паши его привлекают меньше, чем зоны на западе и востоке, то на север и юг он лететь тоже не будет). Тогда, при падении на 2 метра вниз, Паша будет преодолевать 2 метра по горизонтали по направлению ветра. Возникает две ситуации:

1. $H < R$. Паша не успевает долететь до границы (в какую сторону бы ни дул ветер) и падает в джунгли. Ответ - “jungle”.

2. $H \geq R$. Паша успевает долететь до границы и, в зависимости от направления ветра, ответ либо "ruin" (при $W = -1$), либо "forest" (при $W = 1$).

Задача В. Учёные на деревьях

Автор задачи - Владислав Тюльбашев

Задача предполагает умение участника использовать циклы.

В ходе решения считаем и суммируем в одну переменную (назовем её sum) числа из второй строки. Сравним sum и M .

1. Если $M \leq sum$, то общее количество мест на деревьях меньше числа учёных, и те помещаются на деревьях. Рапторы никого не съедят! Ответ: 0.
2. Иначе на деревьях не поместятся $(M - sum)$ ученых. Их съедят. Ответ: $(M - sum)$.

Задача С. Все любят круассаны

Автор задачи - Даниил Сорокин

Задача предполагает знание участником алгоритма Евклида и умение его реализовывать в боевых условиях.

Заметим несколько фактов относительно ответа:

1. Каждый раз, когда первый раптор приходит, кормушка открыта. Это значит, что A делится нацело на T . Действительно, то, что A делится на T , означает, что $A = k * T$, где k - целое. Это значит, что на k -е открытие кормушки (k -е по номеру, а номер - это обязательно целое число вида 1, 2, 3, ...) придёт раптор.
2. Для второго раптора аналогичные рассуждения приведут нас к тому, что B делится нацело на T .
3. Из условия нас просят максимизировать число T (потому что без этого условия задача была бы слишком простой - потому что единица всегда являлась бы одним из возможных ответов).

Сформулированная задача - задача поиска максимального делителя двух чисел, решается алгоритмом Евклида.

Задача D. Черепаха добралась до воды

Автор задачи - Арсений Буланов

Задача предполагает умение участника использовать циклы и одномерные массивы.

В задаче требовалось:

1. Считать два массива длины N .
2. Затем в цикле попарно рассмотреть элементы считанных массивов (итерироваться по i от 1 до N , рассмотреть пары вида $(C[i], R[i])$).
3. Каждая такая пара чисел - скорости черепахи и раптора в одном забеге.
 - а. Если $C[i] < R[i]$, то раптор догонит черепаху (так как раптор стартовал через мгновение после неё, то расстояние до моря смысла не имеет - он поймаёт её практически моментально).
 - б. Если $C[i] \geq R[i]$, то раптор никогда не догонит черепаху - просто потому, что она будет быстрее него, несмотря на то, что он выбежал позже всего лишь на мгновение.
4. Внутри цикла для каждой пары, для которой выполняется случай а, нужно увеличить ответ на единицу.
5. После цикла вывести ответ.

Задача Е. Посторонним В.

Автор задачи - Татьяна Ивановна Грохлина

Задача предполагает наличие воображения и знания признаков делимости числа на 9.

Сразу стоит отметить, что фактически перемножать все числа от 1 до 37 не требуется (значение превышает 10^{43} , считать которое вручную не имеет смысла, а для программного решения необходимы навыки работы с длинной арифметикой (что выходит за рамки Стартовой лиги), либо умение пользоваться Python).

Решим несколько более общую задачу - найдём ответ для случая, когда нужно перемножить не числа от 1 до 37, а от 1 до произвольного N , где $N \geq 6$. Для решения вспомним признак делимости числа на 9 - то, что его сумма цифр делится на 9. Заметим, что для любого $N \geq 6$ в произведение входят числа 3 и 6 ($6 = 2 * 3$), и для любого такого N произведение чисел от 1 до N будет делиться на 9. Далее мы сложим все числа в получившемся произведении, и полученное число тоже будет делиться на 9, мы снова сложим его цифры, и так далее. Таким образом, в конце концов мы дойдём до числа из одной цифры, и оно будет делиться на 9. Такие числа только 0 и 9. Можно несложным образом показать, что из всех чисел 0 может быть суммой цифр только нуля (потому что у любого ненулевого числа есть ненулевая цифра в составе - значит, сумма цифр любого ненулевого числа не является нулём), таким образом, ответ 9.

КЛАССИЧЕСКАЯ ЛИГА

Задача F. Танцы с рапторами

Авторы задачи - Татьяна Ивановна Грохлина и Арсений Буланов

Задача предполагает от участника умение оперировать двумерными массивами и использовать сортировку (собственноручно написанную или стандартную).

Для каждой лужи нужно найти количество рапторов, которых Паша гарантированно успеет порадовать.

1. Сначала найдём номера танцев, которые в теории могут понравиться рапторам (от 00 до 99, номер равен сумме своих цифр + произведению своих цифр). Это 00-09, 19, 29, ..., 89, 99 - суммарно 19 возможных танцев. Паше придётся с каждым раптором танцевать все 19 танцев - только за это количество он гарантированно (то есть, даже если ему максимально не повезёт) найдёт нужный танец.
2. Опять-таки, поскольку нужно гарантировать минимальное количество рапторов, которое сможет порадовать Паша, нужно рассмотреть худший случай для него - а именно, когда ему попадают рапторы с самым долгим минимальным временем танца M_{ij} - сначала раптор с самым долгим минимальным временем танца, потом второй по минимальной длительности танца раптор, и так далее... Для того, чтобы учесть рапторов в таком порядке, их необходимо отсортировать по убыванию внутри каждой лужи - L раз вызвать сортировку от разных строк входной матрицы M .
3. Затем для каждой лужи мы инициализируем переменную sum нулем (она имеет смысл суммы минимальных времён танца рапторов на отрезке от 1 до j) и идем в цикле по j от 1 до R , на каждом шаге проверяем, что если $(sum + M[i,j]) * 19 > T[i] * 60$, то выводим $j - 1$ и переходим к следующей луже, а иначе увеличиваем sum на $M[i, j]$. Если мы ничего не вывели после этого цикла от 1 до R , то это значит, что Паша может найти любимый танец для всех рапторов - в таком случае нужно вывести R . После этого нужно перейти к следующей итерации по i .

Задача G. Великий Рептилоидный Флот

Автор задачи - Даниил Сорокин

Задача предполагает от участника знания задачи о рюкзаке, а также простейшего способа её решения.

Помимо знания этого алгоритма, в задаче не требуется практически ничего. Необходимо запустить этот алгоритм на считаном массиве данных, а также подсчитать сумму массива. Затем, после того, как алгоритм, решающий задачу о рюкзаке, отработает, необходимо пройтись по всем достижимым весам рюкзака, и посмотреть, есть ли среди них такой вес w , что $|w - (\text{sum} - w)| \leq M$.

Задача H. Разбор завалов

Автор задачи - Владислав Тюльбашев

Задача предполагает от участника знания базовых графовых алгоритмов, в частности, DFS.

Решим задачу в терминах графов. Построим граф на $N + 1$ вершинах: для каждой пары (a, b) во входных данных проведём ориентированное ребро от b к a . Теперь посмотрим, какой вид примут некоторые термины из задачи с использованием графового подхода. Если какой-то объект с номером k блокирует Странную Блестящую Машину (вершина с номером 0 в графе), то от 0 к k будет вести ребро. Для того, чтобы убрать объект k , нужно убрать все объекты, которые его блокируют - то есть, все объекты, соответствующие вершинам, в которые ведут ребра из вершины k . Если из них ведут рёбра куда-то, то нужно убрать и эти объекты, и так далее.

То есть, нужно убрать все вершины, которые являются *достижимыми* из вершины 0 , и только их.

Почему это корректно? Потому что для любой достижимой вершины k можно взять путь от 0 до k , и он будет иметь смысл “Объект k блокирует объект k_0 , который блокирует объект k_1 , который ..., который блокирует объект 0 ” - все эти объекты надо будет убрать, чтобы освободить нулевой. С другой стороны, любая недостижимая из 0 вершина k не будет иметь пути из 0 в k и никак не будет влиять на блокировку Странной Блестящей Машины (можно несложно доказать от противного - если бы эта вершина как-то влияла на блокировку, то можно было бы

последовательно брать вершины, которые зависят от неё, и в итоге построить путь от 0 до k, но пути нет по условию - что и требовалось доказать).

Таким образом, нужно подсчитать количество достижимых вершин из вершины с номером 0, это делается алгоритмом DFS с булевым массивом посещённых вершин. Итоговый ответ считается как ((количество посещённых DFS вершин) или же (количество записей true в массиве посещённых вершин)) минус один (потому что DFS пометит ещё и нулевую вершину).

Также можно было решить эту задачу без использования графов. Считая, что в массивах a[] и b[] хранятся начала и концы рёбер, было достаточно написать цикл:

```
for (int i = 0; i < n; i++)  
    for (int j = 0; j < m; j++)  
        if (r[b[j]])  
            r[a[j]] = true;
```

Количество помеченных вершин и будет являться ответом.

Задача I. Инструкция на английском

Автор задачи - Владислав Тюльбашев

Задача предполагает от участника умение работать со строками.

На питоне и с++ задачу можно быстро и безболезненно решить, создав словарь (std::unordered_map), где ключом будет подстрока, а значением - множество (std::set) номеров строк, в которых оно встречалось. Далее, необходимо для каждой строки перебрать все подстроки и занести в словарь с соответствующим номером. После чего достаточно перебрать все ключи в словаре и выбрать самый длинный, у которого размер множества номеров строк, в которых ключ встречается, больше единицы.

ЗОЛОТАЯ ЛИГА

Задача J. Недетские кубики

Автор задачи - Алексей Соловьёв

Задача предполагает от участника знание стандартных графовых алгоритмов и битовых операций над числами. Вероятно, самая сложная задача этой олимпиады.

Представим задачу в виде графа. Вершинами в нём будут все возможные состояния доски размером $N \times M$ (у каждой клетки может быть два состояния - следовательно, размер множества вершин равен $2^{N \cdot M}$). Каждые два состояния, отличающихся между собой только обменом элементов в какой-то паре соседних по стороне клеток, соединим неориентированным ребром стоимости 1. Получится граф. Теперь всё, что нам нужно сделать, - это запустить BFS от вершины, соответствующей стартовому состоянию, и посмотреть расстояние до вершины, соответствующей искомому состоянию.

Как это реализовать? Предлагается каждому состоянию сопоставить битовую маску (развернуть поле $N \times M$ в строку длины $N \cdot M$, потом вместо точки поставить 0, вместо решётки - единицу), а битовую маску интерпретировать как число. Это - взаимно-однозначное преобразование множества полей в подмножество множества целых чисел.

Остаётся разобраться с тем, как генерировать ребра (их нет смысла генерировать отдельно для всех состояний - потому что многие из них окажутся недостижимыми, можно их считать непосредственно в момент обработки соответствующей вершины, так как всё равно каждая вершина будет обработана не более одного раза). Получать по битовой маске рёбра к соседям можно с использованием битовых операций напрямую, а можно применить обратное преобразование и получить по битовой маске поле, затем получить соседние поля и превратить соседние поля обратно в битовые маски, получив список соседей текущей вершины.

Задача К. Даша против

Автор задачи - Владислав Тюльбашев

Задача предполагает от участника либо теории игр Шпрага-Гранди, либо знания о том, как решать задачу об игре в ним.

Эта задача великолепно разобрана в интернете - например, здесь:

http://e-maxx.ru/algo/sprague_grundy

Если XOR-сумма входных чисел из второй строки равна нулю, то ответ "NO", иначе ответ "YES".

Задача Л. Гнев рапторов

Автор задачи - Владислав Тюльбашев

Задача предполагает от участника знания алгоритма Флойда.

Алгоритм можно также посмотреть на e-maxx:

http://e-maxx.ru/algo/floyd_warshall_algorithm

После исполнения алгоритма у нас будут расстояния между всеми парами вершин. Теперь будет достаточно проверить, кто быстрее доберется до развалин лаборатории, - Паша или рапторы. Если Паша успеет добраться, то такую же проверку надо сделать для Даши.