

Разбор задач отборочного тура всеармейского конкурса по программированию «ИнфоТехКвест» (2019-2020гг)

Отбор в сборную

В данной задаче требовалось определить фамилию кадета самого среднего роста. Во избежание неоднозначностей, по условию задачи всегда предлагалось нечётное количество кадет.

Поскольку выбирать кадета требуется по росту, а выводить его фамилию, для решения этой задачи необходимо использовать тип данных с двумя полями, т.е. структуру или класс, в зависимости от возможностей используемого языка программирования. Созданный тип данных должен содержать фамилию кадета и его рост, причём рост всех кадет необходимо хранить в одних и тех же единицах измерения. Поскольку для перевода всех единиц измерения в одну в условии задачи приведены миллиметры, целесообразно хранить рост каждого кадета именно в миллиметрах.

Необходимо создать массив объектов типа «кадет» заданного размера и в каждый элемент массива ввести его фамилию и рост, переведённый в миллиметры. Для этого необходимо в зависимости от введённой единицы измерения введённый рост умножить на соответствующий этой единице измерения коэффициент. После этого необходимо этот массив отсортировать по росту и вывести фамилию кадета, оказавшегося в середине массива.

Переливание крови

В данной задаче предлагалось, опираясь на количество крови каждой группы и резус-фактора в банке крови, определить, сколько крови какой группы и резус-фактора перелить человеку, которому требуется перелить определённое количество крови. Данная задача разбивается на следующие подзадачи:

- Для каждой группы крови и резус-фактора сформировать упорядоченный список подходящих кровей других групп или резус-факторов. Например, для первой группой крови и отрицательным резус фактором этот список будет пустым, а для четвёртой группы крови и положительного резус-фактора он будет выглядеть следующим образом: IV-, II+, III+, I+, II-, III-, I-. Для удобства можно полное описание крови (группа и резус-фактор) представить в виде перечислимого типа.
- Проверка общего количества крови в банке крови, подходящего человеку. Это сумма количества такой же крови, как и у реципиента, и количеств кровей в банке, которые ему можно перелить. Если общего количества подходящей крови в банке недостаточно, по условию задачи, переливание крови не производится. Если возможно, то выполняется следующий шаг.
- Обходится список подходящих кровей в банке. Для каждой из них выполняются следующие действия:
 - Если количество крови, которое требуется, больше, чем крови определённой группы и резус-фактора в банке, вся эта кровь переливается реципиенту. При этом количество соответствующей крови в банке обнуляется, а количество требуемой крови для реципиента уменьшается на эту величину.
 - Если количество требуемой крови не превышает количество проверяемой крови в банке, то количество крови в банке уменьшается на количество требуемой крови, и цикл прерывается.
- Вывод того, что после всех этих манипуляций осталось в банке крови.

Кто кого переговорит

В данной задаче требовалось в зависимости от исходных данных выбрать одно из двух решений: либо говорить больше слов в минуту, чем все кадеты в классе, либо выгнать самого болтливую. В то время, как первый вариант решается элементарным суммированием количества слов, произносимых всеми кадетами, с последующим сравнением с

возможностями преподавателя и добавлением единицы, то второй вариант требует дополнительных замечаний.

Дело в том, что болтливость каждого кадета складывается из двух компонент: количества слов, которые произносит он сам, и количества слов, которые говорят ему. Поэтому каждый раз при вводе информации, кто кому и сколько слов в минуту говорит, необходимо для каждого кадета, участвующего в диалогах, увеличивать количество произносимых слов. Исключение составляют только те кадеты, которые говорят сами с собой. Как и в случае с задачей «Отбор в сборную», для хранения этой информации целесообразно использовать специализированный тип данных с двумя полями – фамилия кадета и его суммарная болтливость. Для выбора самого болтливого кадета целесообразно завести переменную с номером самого первого кадета. В начале им считается самый первый кадет. Затем надо обойти всех кадет, кроме самого первого, и сравнивать болтливость кадета, выбранного в качестве самого болтливого, с этим номером, и если очередной кадет окажется болтливее, номер надо заменить. После перебора всех кадет остаётся вывести фамилию кадета, который оказался самым болтливым.

Шифрование

В данной задаче требовалось посчитать количество неупорядоченных пар различных простых чисел, которые больше или равны l но меньше или равны r . Данная задача очевидным образом разбивается на 2 подзадачи:

- Найти количество простых чисел в данном диапазоне (очевидно, что количество пар зависит только от количества простых чисел)
 - Зная количество простых чисел определить количество пар
- Решить первую подзадачу можно было 2 способами:
1. Проверить каждое число x в отрезке $[l, r]$ на простоту попробовав разделить каждое число на все натуральные числа отрезке $[2, \sqrt{x}]$. Если число не делится ни на одно из этих чисел, то оно простое.
 2. Построить решето Эратосфена для чисел до 10^5 . Результатом будет массив a , такой, что $a[i]=1$ если i – простое, а иначе $a[i]=0$.

Пусть количество простых чисел k . Посчитать количество пар можно размышляя следующим образом: образуя пару мы выбираем два элемента. Первый мы выбираем k способами, а второй $k-1$ способом (поскольку один из элементов мы уже выбрали в качестве первого). Осталось учесть, что каждую пару мы посчитали 2 раза (пары (p, q) и (q, p) на самом деле одинаковые, хотя мы почитали их двумя различными). В итоге формула выглядит следующим образом: $k*(k-1)/2$

Обмен книгами

Можно заметить, что когда мы считаем ответ для ребенка i , мы также считаем ответ для детей $p[i]$, $p[p[i]]$ и так далее. Это позволяет значительно уменьшить время нахождения ответа. Таким образом решение выглядит следующим образом:

- Обходим всех детей по очереди (понятно, что если для ребенка ответ уже посчитан то мы его пропускаем)
- Выбрав конкретного ребенка будем отслеживать номера детей, к которым попала книга этого ребенка.
- Как только книга первый раз попадет обратно к выбранному ребенку, запишем ответ для него.
- Из сказанного в самом начале следует, что такой же ответ будет и тех детей, к которым попала книга выбранного ребенка. Сразу запишем ответ и для них тоже.

Дежурство по автопарку

В данной задаче требовалось имитировать действия дежурного по автопарку, регистрирующего въезжающие и выезжающие машины. В момент приёма наряда дежурный узнаёт, какие машины находятся в парке, во время наряда он регистрирует въезжающие и

выезжающие машины, при сдаче наряда он должен предоставить информацию о находящихся в парке машинах.

Поскольку в ходе работы автопарка список машин может расширяться за счёт машин, приезжающих в автопарк, для хранения данных о машинах целесообразно использовать список. Поскольку машины, приезжающие в автопарк впервые, встречаются лишь эпизодически, а обращаться к машине, расположенной в определённом боксе, требуется при обработке каждой команды, предпочтительным является линейный, а не связный список. Кроме того, для поиска машины в автопарке по её номеру, следует создать карту соответствия номера машины номеру бокса, в котором находится машина. Тогда список машин может представлять список объектов специального типа с двумя полями: номер машины и логический признак, находится ли она в автопарке.

Дальнейшая программа представляет собой цикл, в котором вводится и выполняется команда, продолжающийся до тех пор, пока команда не будет называться `delivery`.

Если команда начинается со слова `in`, необходимо преобразовать остаток команды в номер машины и проверить, есть ли уже машина, которая нас интересует, в карте соответствия номеров машин номерам боксов. Если она есть, нужно обратиться к этой машине по номеру бокса, найденного в этой карте соответствия, и проверить, где она располагается. Если она не в парке, необходимо пометить, что она в парке. В противном случае, нужно завершить программу с сообщением об ошибке.

Если команда начинается со слова `out`, нужно из остатка команды получить номер машины и найти в автопарке номер её бокса по карте соответствия. Если он отсутствует, либо соответствующая машина помечена как отсутствующая в автопарке, необходимо вывести сообщение об ошибке. В противном случае надо пометить машину как отсутствующую в автопарке.

После цикла по командам необходимо перебрать все машины, в списке машин, и если машина находится в автопарке, вывести её номер.

Тактика

В данной задаче требовалось увидеть, сколько противников окажутся в области прямой видимости с бойцом при наличии заграждений. В условии задачи заграждения расположены параллельно и перпендикулярно осям сетки координат. Решение данной задачи включает в себя следующие подзадачи:

- Перебрать всех противников на поле в цикле.
- По координатам бойца и его противников вывести уравнения прямых, соединяющих бойца и с каждым противником. Если принять уравнение искомой прямой за $y = a \cdot x + b$, то a будет равно $(y_1 - y) / (x_1 - x)$, а b будет равно $y - x \cdot a$.
- Найти пересечения отрезка прямой от x до x_1 со всеми заграждениями. Если таковых не найдено, боец находится на расстоянии прямой видимости с противником. Для этого нужно в уравнение линии, соединяющей бойца с противником, нужно подставить все четыре координаты прямоугольника и посмотреть, попадает ли вторая координата в границы прямоугольника. Если хотя бы одна координата попадает, значит, заграждение преграждает прямую видимость. При этом рассматривать нужно только те заграждения, правая граница которых правее левой координаты отрезка, соединяющего бойца и противника, а левая левее правой координаты этого отрезка.
- Подсчитать количество отрезков, не пересекаемых заграждениями.

Если координаты x и y бойца и противника совпадают, этот метод применить не получится.

Уравнение прямой будет иметь вид $x = x_1$, и потребуется просто проверять координаты границ заграждений, сравнивая их с x_1 и y координатами бойца и противника. Если x_1 находится между левой и правой сторонами заграждения, а y координаты заграждения находятся между y координатами бойца и противника, то заграждение препятствует прямой видимости, в противном случае нет.

Морской бой

В данной задаче требовалось определить, в каком направлении требуется выпустить торпеду с подводной лодки, если скорость и направление движения цели засечены по двум точкам, известны скорость торпеды и время, необходимое на её подготовку. Эта задача делится на следующие подзадачи.

- Определение скорости цели по x , по y и суммарной (считается по теореме Пифагора как корень из суммы квадратов скоростей по x и по y).
- По этим скоростям находим, где будет находиться цель в тот момент, когда торпеда будет готова к запуску. Для этого прибавим к координатам цели во второй точке засечки скорости по соответствующим координатам, умноженные на время подготовки торпеды. Заметим, что поскольку нас интересует только направление, можно для упрощения расчётов вычесть из них координаты подводной лодки, поместив ее, таким образом, в начало координат.
- Найдём угол между направлением движения цели и направлением от подводной лодки на цель. Заметим, что направление от подводной лодки на цель задаётся как вектор из координат цели в момент выстрела, а направление движения цели - как вектор её скоростей по x и по y . Поделив скалярное произведение этих векторов на произведение их длин, найденных по теореме Пифагора, получаем косинус искомого угла, и применяем функцию арккосинуса. Для решения задачи нам понадобится внешний угол по отношению к найденному.
- Рассмотрим треугольник, образованный отрезком между подводной лодкой и целью в момент выстрела, и точкой, в которую необходимо целиться, чтобы попасть. Один из углов мы узнали на предыдущем шаге. Противлежащая ему сторона и сторона, по которой будет двигаться цель, должны соотноситься так же, как и скорости цели и торпеды соответственно. По теореме синусов получаем, что синус угла между направлением на цель в момент выстрела и направлением, куда надо целиться, равен произведению синуса угла, найденного на предыдущем шаге, на скорость цели, делённому на скорость торпеды.
- Сравнив направления векторов, определяем, что делать с полученным углом. Если вектор направления движения цели находится по часовой стрелке от направления на цель в момент выстрела, необходимо прибавить этот угол к азимуту направления на цель. Если против часовой стрелки, то отнять.
- Не забываем, что азимут должен находиться в диапазоне от 0 до 360 градусов не включительно, и при необходимости либо прибавляем, либо отнимаем от полученного числа 360.

Это решение имеет вырожденный случай, когда направление на цель в момент выстрела совпадает с направлением движения цели. В этой ситуации азимут выстрела должен просто совпадать с азимутом направления на цель в момент выстрела.

Морской бой (альтернативные решения)

Выше было приведено численное решение задачи "Морской бой". Оно является наиболее точным и эффективным с точки зрения производительности, но в то же время наиболее сложно в реализации. Однако данную задачу можно решать и численными методами, и хотя они более ресурсоёмкие, такие решения всё равно будут укладываться в предложенные временные нормативы. Принципиально решений может быть два:

- Поскольку в задаче предлагается определить азимут с точностью до градуса, можно перебрать все 360 градусов и выбрать наилучший
- Можно использовать метод пристрелки: смотреть, где окажется цель, если стрелять без упреждения, и итеративно корректировать азимут.

Вне зависимости от метода решения задачи, первые шаги аналитического решения всё равно остаются: необходимо найти скорость движения цели по x и y и её координаты в момент выстрела относительно подводной лодки. После этого при решении задачи методом пристрелки требуется сделать следующее:

- Построить прямоугольный треугольник, гипотенуза которого соединяет начало координат (положение подводной лодки) с целью. По теореме Пифагора по координатам цели найти длину гипотенузы.
- Полученная гипотенуза - это длина пути торпеды. Зная скорость движения торпеды, можно определить время, за которое торпеда пройдёт этот путь. Зная это время, координаты цели в момент выстрела и её скорости по x и y , рассчитываем новые координаты цели, которые она будет иметь в момент попадания торпеды в то место, где была цель в момент выстрела.
- Повторяем эти расчёты для новых координат цели до тех пор, пока разница между двумя последними результатами превышает заданную погрешность (в нашем случае - 1 градус). По поводу этого решения можно сделать такое замечание. Если угол между вектором движения цели и направлением от подводной лодки на цель в момент выстрела получается тупой или хотя бы прямой угол, ряд приближенных решений будет сходиться к точке на линии движения цели, асимптотически к ней приближаясь, но никогда не достигая. Поэтому в прикладной ситуации даже узкий диапазон углов может дать большую погрешность в расстоянии, и торпеда в цель не попадёт, и оценивать нужно именно разность координат. Что касается решения прямым перебором, то для каждого из азимутов необходимо:
 - Посчитать, какая получается скорость по x и y торпеды, если её выпустить с этим азимутом.
 - Посчитать, за какое время она поравняется с целью по x и по y . Это можно сделать по таким формулам:
 - $t_x = x_3 / (v_{tx} - v_{cx})$
 - $t_y = y_3 / (v_{ty} - v_{cy})$,
 - где t_x и t_y - это искомое время, x_3 и y_3 - координаты цели в момент выстрела, v_t и v_c - скорости торпеды по обеим координатам.
 - Отдельно следует обработать вырожденные случаи.
 - Если знаменатель в одной из формул получился равным 0, это означает, что при данном азимуте разность по соответствующей координате не меняется. Попасть в цель в данной ситуации можно только тогда, когда данная координата у подводной лодки и цели совпадает изначально.
 - Если оба знаменателя равны 0, значит, торпеда и цель двигаются параллельно, и цель никогда не будет поражена.
 - Если время, за которое торпеда поравнялась с целью по одной из координат является отрицательным, это означает, что стрелять надо было в противоположную сторону.
 - Среди всех азимутов следует выбрать тот, при котором модуль разности времён, рассчитанных по указанным формулам, минимальна. Если при каком-то из азимутов она получилась равно 0, на этом можно прерывать перебор - найдено точное решение задачи.