Разбор задач городского тура олимпиады по информатике

Общие замечания

В данном тексте описаны наиболее важные соображения для решения каждой из задач. В архиве находятся подробно прокомментированные решения задач (на Delphi).

Задача А. Три стрелы

Расстояние, которое пролетит стрела, пропорционально синусу удвоенного угла, под которым она была выпущена (в курсе физики эта формула приводится в теме «Движение тела, брошенного под углом к горизонту»). Все остальные параметры (начальная скорость и начальная точка) для стрел одинаковы. Поэтому вполне достаточно сравнивать синусы удвоенных углов.

Заметим, что можно обойтись и без вычисления синусов. Во-первых, известно, что $\sin\alpha = \sin(\pi - \alpha)$, $\alpha \le \pi/2$ (= 90^0). Поскольку углы во входном файле не превосходят 90^0, удвоенные углы, соответственно, не превосходят 180^0. Следовательно, $\sin 2\alpha$ можно заменить $\sin 2\beta = \sin(\pi - 2\cdot(\pi/2 - \beta))$, $\alpha = \pi/2 - \beta$. На отрезке от 0 до $\pi/4$ функция $\sin 2\alpha$ ($\sin 2\beta$) является возрастающей. Поэтому, если угол, под которым выпустили стрелу, превосходит 45^0, заменим его углом, меньшим 45^0 (согласно формуле, приведенной выше). Теперь достаточно отсортировать полученные числа в порядке возрастания с учетом старшинства братьев.

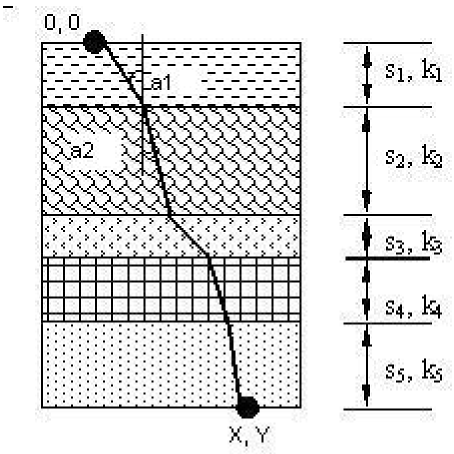
Задача B. Высокие горы

Как можно видеть, тактика, состоящая в том, что обоз должен двигаться, пока хватает сил, а отдыхать ровно столько, чтобы хватило до следующей точки «перевала», как раз и приведет к минимальным затратам на время отдыха.

Последовательно анализируются очередные точки «перевала», до которых надо добраться, на предмет того, достаточно ли на это у обоза сил. Если достаточно – обоз двигается к этой точке. Если же нет, необходимо еще проверить, можно ли в принципе набрать достаточный запас сил – т.е. не превосходят ли требуемые затраты величины E_{max}. В случае, если превосходят, можно прерывать рассмотрение точек и выводить ответ NO. Еще один момент, на который следует обратить внимание – для величины V число 0 является допустимым значением (обоз не может восстановить силы в принципе). Такой случай лучше рассмотреть отдельно.

Задача C. Широкие поля.

В этой задаче довольно несложно увидеть аналогию с движением светового луча в многослойной среде. Показатель преломления при этом сопоставляется величине, обратной «коэффициенту проходимости». Согласно закону преломления света $\sin\alpha_1/\sin\alpha_2 = n_2/n_1 = k_1/k_2$. Вероятно, наиболее рациональным способом решения является приведение всех углов к одному (например, углу, под которым должен двигаться



всадник в наиболее «плотной» среде – т.е. с наибольшим коэффициентом проходимости; все остальные углы будут заведомо больше него). Отыскать же нужный угол (а точнее, синус нужного угла – непосредственно угол на самом деле вычислять и не нужно) можно с помощью бинарного поиска – разделяя каждый раз диапазон возможных значений пополам и сокращая область поиска.

Когда наиболее короткий путь будет найден, вычислить время, требуемое на его преодоление, представляется уже чисто технической задачей.

Задача D. Дремучие леса

Один из возможных способов решения задачи – эмуляция действий волка и Ивана-царевича. Условно говоря, система «волк – Иван-царевич» может находиться в трех состояниях:

1. волк везет Ивана-царевича
2. волк отдыхает, Иван-царевич идет самостоятельно
3. волк догоняет Ивана-царевича.

Поэтому задача сводится к описанию переходов из одного состояния в другое с учетом переходов с одного участка на другой (переходы приводят к изменению скоростей движения). Разумеется, надо учитывать, что, когда волк догоняет Ивана-царевича, он расходует силы, и сможет везти его в течение времени, меньшего G. Следует также обратить внимание на то, что в задаче требуется узнать, за какое время до болот доберется Иван-царевич. Волк за это время может до болот и не добраться.

Задача E. Купеческие дочери.

Данную задачу можно решать, например, методом динамического программирования. Идея заключается в том, чтобы искать стоимости наборов украшений, номера которых не превосходят некоторого значения j, и постепенно увеличивать значение j. В этом случае можно составить матрицу, в которой номер строки указывает верхнюю границу номеров рассматриваемого подмножества предметов, которые включаются в первый набор. Номера столбцов (k) соответствуют суммарной стоимости включенных в набор драгоценностей. В ячейках же записывается разница в стоимости между первым и вторым (полным комплектом за вычетом включенных в первый) набором драгоценностей.

Если предмет №j в набор не включается, то задача сводится к задаче меньшего размера - с верхней границей j-1. Обозначим через f1 имеющуюся стоимость первого набора, тогда стоимость второго f2 = FullCost – f1 (где FullCost – полная цена всего набора). Если некоторое украшение стоимостью cj перекладывается из первого во второй набор, то разность в стоимостях f1 – f2 = 2*f1 – FullCost изменится, и станет равной f1 + cj – (f2 - cj) = f1 + 2cj – (FullCost – f1) = 2*f1 + 2*cj – FullCost.

Поэтому можно использовать следующую рекуррентную формулу:

costs[j, k] := min(costs[j-1, k], (costs[j-1, k-c[j]] + 2*c[j])),

где минимум вычисляется по абсолютному значению разницы стоимостей набора без «последнего» элемента (№ j) и набора с учетом этого элемента. Разумеется, формулу можно применять лишь при k >= c[j].

Задача F. На болоте.

Задача состоит в отыскании кратчайшего пути в графе от вершины № 1 до вершины № N. Для этого можно воспользоваться, например, алгоритмом Дейкстры. Его описание можно найти во многих источниках – как в книгах, так и в сети Internet. Здесь кратко изложим его основные положения.

Заметим, что алгоритм Дейкстры подходит только для графов, у которых все ребра имеют неотрицательный вес (в отличие, например, от алгоритма Форда-Беллмана). Все множество кочек вначале разделим на два подмножества, которые назовем выделенными (т.е. проанализированными) и невыделенными. Изначально выделенной является только одна – первая – кочка.

Для каждой кочки хранится (в массиве) путь минимальной длины до нее от первой кочки, проходящий только через выделенные кочки. Все значения в массиве (за исключением, быть может, первого) следует инициализировать числом, заведомо превосходящим длину самого длинного пути в графе.

На первом шаге некоторые значения в массиве обновятся – фактически в него будут внесены пути до кочек, на которые можно попасть непосредственно с первой кочки. Теперь из всех имеющихся путей выбирается минимальный, и кочка, соответствующая ему, включается во множество выделенных кочек. Дальнейшие действия таковы: рассматриваются кочки, непосредственно достижимые с только что выделенной. Для них сравниваются пути, проходящие через выделенную кочку, и пути, уже хранящиеся в массиве. Выбирается наименьший из них и, при необходимости, запись в массиве обновляется. Затем среди невыделенных кочек выбирается та, путь до которой самый короткий, и включается во множество выделенных.

Поскольку гарантируется, что путь до кочки с номером N всегда существует, она рано или поздно попадет во множество выделенных кочек. Вообще говоря, после этого можно прерывать вычисления.

Задача G. Знакомство.

Общее время сна вычисляется как сумма

а) быстрой фазы сна – царевна-лягушка ждет начала медленной фазы, чтобы превратиться в Василису Премудрую б) времени, которое будет затрачено на путешествие – отношение S / V (с ним тоже надо обойтись аккуратно, поскольку при делении может получиться ненулевой остаток)

в) если приземление кареты произошло в течение быстрой фазы сна, – времени до начала очередной медленной фазы плюс собственно эта медленная фаза; если же приземление произошло во время (или строго в момент окончания) медленной фазы, то времени до ее окончания

г) времени последней быстрой фазы сна.

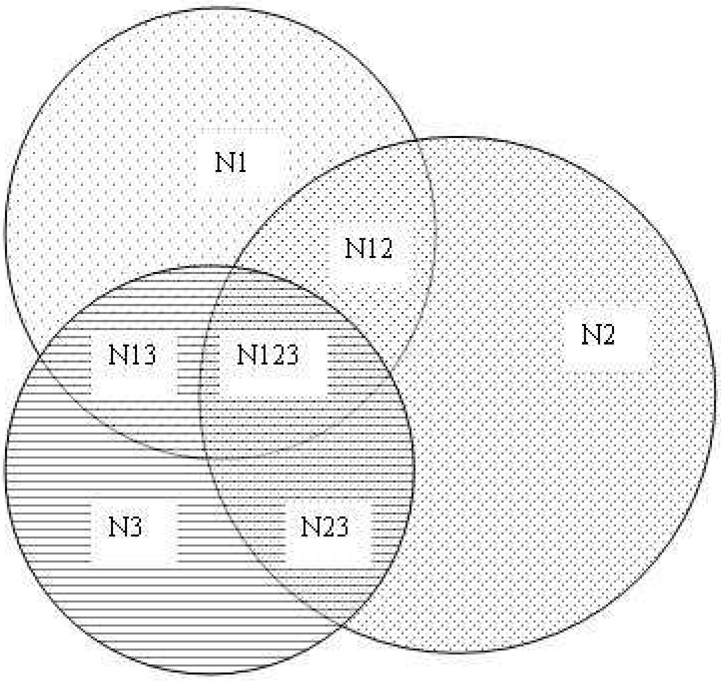
Задача H. Приданое.

В этой задаче для перекладывания вещей важны только первые три цифры инвентарного номера (с ними можно работать и как с целым числом, и как с подстрокой). Наиболее простым путем является пошаговое рассмотрение содержимого каждой из телег, которое повторяется как минимум два раза (при самом удачном стечении обстоятельств все вещи обретут свое место еще на первом проходе, но нужно убедиться, что больше ничего перемещать не требуется). (Обратите внимание: просто сортировать вещи нельзя!)

Как происходит перемещение – подробно иллюстрирует пример, приведенный в условии задачи.

Задача I. Гости дорогие.

Общее число гостей должно вычисляться как сумма тех частей кругов, которые не имеют общих точек с другими кругами, и «трилистника», образованного всеми пересечениями.



Часть первого круга – число гостей, которых пригласил только старший брат – может быть вычислена как N1 – (N12 + N13 – N123). Аналогичным образом посчитаем гостей, которых пригласил только средний брат: N2 – (N12 + N23 – N123), и тех, которых пригласил только младший брат: N3 – (N13 + N23 – N123). Площадь «трилистника», в свою очередь, вычисляется как сумма частей, которые не имеют общих точек и их пересечения: (N12 – N123) + (N23 – N123) + (N13 – N123) + N123. Объединяя эти формулы, получаем:

$$N = N1 - (N12 + N13 - N123) + N2 - (N12 + N23 - N123) + N3 - (N13 + N23 - N123) +$$
$$+ (N12 - N123) + (N23 - N123) + (N13 - N123) + N123.$$

Раскрывая скобки, приходим к окончательной формуле:

$$N = N1 + N2 + N3 - N12 - N12 - N23 - N13 + N123$$