Разбор задач 1-го тура муниципального этапа

Всероссийской олимпиады школьников по информатике 2011 / 2012 учебного года

Задача A. Правильное управление.

В этой задаче надо выбросить лексикографически минимальным способом M-N+1 ребро так, чтобы от графа осталось только дерево.

Для начала поймем, что если ребра отсортированы в лексикографическом порядке, то мы должны выбросить ребро (u,v) тогда и только тогда, когда по ребрам позже него из u достижима v. Действительно, если по остальным ребрам достижимость нарушается, то мы не можем не брать это ребро. Если путь из u в v использует хотя бы одно оставленное раннее ребро, то ответ неоптимален (добавим ребро (u,v), получим цикл, удалим из него любое другое ребро).

Осталось для каждого ребра проверить указанное выше соотношение. Это можно сделать, перебирая ребра в обратном лексикографическому порядке, при этом у нас должна быть некоторая структура данных, умеющая поддерживать компоненты связности графа после добавления ребра. Самая простая из них - лес непересекающихся множеств. Или можно сделать то же самое, храня для каждой вершины номер ее компоненты связности, при этом при добавлении ребра перенумеровывая только меньшую из объединяющихся компоненту связности за O(ее размера).

Примечание. В самом деле, можно показать, что дополнение ответа данной задачи до множества всех ребер, записанное в обратном порядке, есть лексикографически максимальное остовное дерево, а описанное выше решение - алгоритм Крускала, вес ребра в котором определяется лексикографическим порядком ребра.

Задача B. Удачный выбор.

Все, что требовалось в задаче, - отыскать “локальный минимум” в последовательности. Конечно, он может не совпадать с “глобальным” минимумом последовательности. Для этого достаточно выполнить один проход по ней, последовательно анализируя каждую тройку трансформерпадов. Хранить все цены нет необходимости, достаточно запоминать наименьшую цену, удовлетворяющую описанию в условии.

Задача C. Важное поручение.

Несложно выписать все возможные различные пути Василия – их всего 13. Для каждого из них можно посчитать затраченное время, а затем выбрать минимум. Также можно написать переборное (рекурсивное) решение.

Задача D. Легкий путь.

В этой задаче необходимо сначала построить граф, ребра которого определяются как кратчайшие расстояния между прямоугольниками. Чтобы посчитать расстояние между двумя прямоугольниками, найдем разности между X-координатами “более левой” из правых сторон и “более правой” из левых, а также между Y-координатами “нижней из верхних сторон” и “верхней из нижних сторон”. Если обе разности положительные, то кратчайший путь между прямоугольниками соединяет их угловые точки. Если же только одна из разностей является положительной, то кратчайший путь – отрезок, параллельный соответствующей оси координат. Наконец, если обе разности нулевые или отрицательные, то прямоугольники перекрываются (либо касаются друг друга). В этом случае кратчайшее расстояние между ними равно нулю. Также следует посчитать расстояния от точек старта и финиша до всех прямоугольников.

Теперь, когда граф построен, можно отыскать кратчайший путь из точки старта до точки финиша, например, алгоритмом Дейкстры. Заведем массив d[i] оценок расстояний от точки 0 (точки старта) до всех остальных точек. Сначала можно считать, что мы попадаем в каждую точку непосредственно из точки 0, а затем действовать следующим образом. Выберем точку, для которой оценка самая маленькая – пусть это точка k. Поскольку расстояния между прямоугольниками неотрицательны, мы уже не доберемся до нее более коротким путем. Однако имеет смысл рассмотреть пути до других точек: вполне возможно, что для каких-то j ($j \neq 0$, $j \neq k$) окажется, что d[j] > d[k] + dist(k,j), где dist(k,j) – расстояние между точками k и j. В таком случае оценка может быть заменена. Пересчитав все оценки, мы можем повторить процедуру, выбрав из вновь полученных оценок наилучшую.

Задача E. Сложная зависимость.

Для каждого зомби посчитаем момент времени, в который он дошел бы до дынепульты, если бы в него не стреляли. Как понятно, это будет время его старта + 11 (он должен сделать 10.5 шагов)

Теперь надо посчитать, в какой момент какого из зомби мы сможем дезориентировать. В первого стреляем в момент, равный времени его старта. Дальше делаем следующее – прибавляем ко времени последнего запуска снаряда (обозначим его lastFire) 3 с и сравниваем:

а) если это время меньше времени старта следующего зомби, то делаем lastFire равным времени его старта

б) если это время больше или равно времени старта, но меньше времени финиша следующего зомби, то оставляем lastFire как есть

в) если это время больше времени финиша – то зомби удастся добраться до оператора дынепульты.