Задачи по теории алгоритмов от Н.К. Верещагина

Если Вам понравилась одна из этих задачи, и Вы хотели бы её решать, обращайтесь к проф. кафедры математической логики и теории алгоритмов Н.К. Верещагину nikolay.vereshchagin@gmail.com.

Задача о кошках и муравьях

Дан некоторый ориентированный граф с выделенной вершиной s и натуральные числа k, n. По ребрам графа (в указанном на ребре направлении) могут двигаться кошки и муравьи. Количество кошек равно k, а муравьи рассматриваются, как непрерывная субстанция. Сначала все муравьи и кошки находятся в вершине s. Затем муравьи и кошки по очереди делают ходы. За один ход любое количество кошек может сдвинуться по ребрам графа на любое расстояние (но только в указанном на ребрах направлении); разные кошки могут перемещаться в разные вершины. То же разрешено и муравьям: за один ход любое количество муравьев может сдвинуться по ребрам графа на любое расстояние (но только в указанном на ребрах направлении). Муравьи и кошки образуют две единых команды и каждой команде доступна информация о положении всех кошек и муравьев (то есть, речь идет об игре с полной информацией). Цель кошек состоит в том, чтобы после каждого их хода в любой вершине графа, в которой находится не менее $1/n$-ой всех муравьев, сидела хотя бы одна кошка.

Вопрос: при каком наименьшем k у кошек есть выигрышная стратегия в описанной игре?

Разумеется, ответ на этот вопрос зависит от графа, но можно показать, что для любого графа достаточно иметь $k = O(n^2)$ кошек. Вопрос: существует ли граф (возможно бесконечный), для которого необходимо $\Omega(n^2)$ кошек?

Задача об избегании окрашенных мест

Пусть даны натуральные числа n, m, k. Рассмотрим следующую игру с полной информацией двух игроков, Алисы и Боба. В начале игры в некоторой клетке бесконечной во все стороны клетчатой доски стоит фишка. Своим первых ходом Алиса может испачкать n клеток на доске. После этого игроки ходят по очереди. В свою очередь каждый из игроков может спасовать (ничего не делать) или сделать ход, то есть, сдвинуть фишку на соседнюю клетку справа или вверху. Алиса стремится поставить фишку на испачканное поле, а Боб — на чистое. Алиса может сделать не более m ходов, а Боб — не более k ходов (количество пасов не ограничено). Игра заканчивается, когда оба игрока спасовали (например, из-за того что лимит его ходов исчерпан). Если в конце игры фишка оказалась на грязном поле, Алиса выиграла, а иначе — Боб.

По теореме Цермело, в этой игре (при любых n, m, k) либо Алиса имеет выигрышную стратегию, либо Боб. Например, при $n = 0$ выиграет Боб.

Вопрос: при каких n, m, k выиграет Алиса, а при каких — Боб?

Задача о разделении ресурсов в иерархической структуре

Дано конечное дерево с корнем. Вершины этого дерева мы будем понимать как подразделения некоторого учреждения: корень — само учреждение, его сыновья — отделы в учреждении, их сыновья — подотделы отделов и так далее. Каждый из отделов генерирует запросы на пространство, каждый запрос есть неотрицательное рациональное число. (Можно представлять себе, что речь идет о памяти на твердом диске.) При этом сумма запросов сыновей любой вершины v не должна превышать запроса самой вершины v. Запрос корня дерева не меняется и равен 1. Запросы же всех остальных вершин могут со временем увеличиваться (при условии сохранения основного требования про сумму запросов сыновей). Уменьшаться запросы не могут, в начальный момент все запросы, кроме запроса корня, равны нулю.

Менеджером памяти называется алгоритм, который, получая запросы, распределяет пространство. Пространство, доступное менеджеру — это интервал $(0, c)$ действительной оси и все это пространство считается принадлежащим корню дерева. Пространство, принадлежащее остальным вершинам, распределяется Менеджером, который в своих действиях ограничен тремя условиями: (1) пространством, которым владеет вершина v, должен владеть и отец v, (2) пространства, которым владеют разные сыновья v, должны не пересекаться и (3) если r есть текущий запрос некоторой вершины, то эта вершина должна владеть некоторым интервалом (a, b) длины r. В каждый момент времени (скажем день в году) Менеджер узнает все текущие запросы и выдает пространство вершинам по своему усмотрению. Выданное пространство отбирать назад нельзя.

Вопрос: при каком значении c (длина всего доступного пространства) у Менеджера есть стратегия действий, удовлетворяющая всем трем требованиям.

Разумеется, ответ на этот вопрос зависит от данного дерева. Можно показать, что для любого графа глубины d у Менеджера есть стратегия для $c = O(d)$. Вопрос: а каково минимально возможное c для бинарного дерева глубины d?

Задача об он-лайн паросочетаниях

Пусть дан конечный двудольный граф и натуральное число m. Множество вершин левой доли графа будем обозначать через L, а множество вершин правой доли — через R. Представим, что нам по очереди дают m различных вершин $x_1, \ldots, x_m \in L$ из левой доли графа. После получения каждой вершины x_i мы должны указать для нее вершину y_i правой доли так, чтобы в результате получилось паросочетание. Последнее означает, что все вершины $y_1, \ldots, y_m$ попарно различны и для всех $i \le m$ вершина y_i соединена ребром с вершиной x_i. Если это возможно (для любой последовательности из m различных левых вершин), то говорят, что граф *допускает он-лайн паросочетание вплоть до m*. Тут важно, что мы должны указать пару для x_i, не зная, каковы будут вершины $x_{i+1}, \ldots, x_m$.

Например, полный двудольный граф, у которого в правой доле не меньше m вершин, допускает допускает он-лайн паросочетание вплоть до m. Нас интересует, есть ли такой граф, в котором правых вершин $O(m)$, и при этом любая левая вершина имеет не слишком много соседей.

Точнее, надо ответить на следующий вопрос: верно ли, что для всех n, m существует двудольный граф с $|L| = n$, $|R| = O(m)$, допускающий он-лайн паросочетания вплоть до m, причем любая левая вершина имеет не более $O(\log n)$ соседей?

Можно доказать, что если количество соседей увеличить до $O(\log^2 n)$, то такой граф существует.

Пояснение. Двудольный граф задается парой множеств L, R и некоторым подмножеством $E \subset L \times R$. Элементы множества L называются левыми вершинами, а элементы множества R — правыми вершинами. Пары из E называются ребрами и говорят, что левая вершина u соединена с правой вершиной v, если пара (u, v) является ребром графа.