Теория вероятностей

Основы

- Интуитивные определения вероятностей: подбрасывание монетки, стрелок, брак на заводе
- Правило сложения вероятностей, несовместные события, пример дартса
- Переход от событий к случайным величинам
- Математическое ожидание — определение на основе частот событий и переход в пределе к вероятностям; стрелок в тире на очки
- Простые задачи на вероятность, сводящиеся к комбинаторике и подсчёту случаев (когда в мешке лежат заданные буквы, достают по одной, нужно подсчитать вероятность собрать заданное слово)
- Вложенный рандом: есть функция $\mathrm{rand}(n)$, которая возвращает целое число от 0 до $n-1$, равновероятно. Эти randы вкладываются друг в друга k раз. Требуется найти вероятность того, что в результате получится заданное число a (простая квадратичная динамика)
- Парадокс дней рождения: по заданному числу дней в году и заданным процентам p найти такое минимальное количество людей, чтобы вероятность того, что у двух из них день рождения совпал, была $\geq p\%$ (простая итеративная формула: начинаем с $p[1]=1$, и потом итерируем)
- Команда играет n игр, каждую выигрывает с вероятностью p. Найти мат. ожидание длиннейшей серии побед (динамика за куб с состоянием d[games][cur_serie])
- Играют двое, у каждого есть свой кубик, на сторонах которого написаны какие-то заданные стоимости, и известно, сколько раз каждый игрок должен кинуть кубик. Стоимости до 1000, бросков до 20. Найти вероятность того, что первый победит второго (сначала посчитаем отдельно для каждого игрока для каждой суммы вероятность её получить, потом пройдёмся циклом)

Циклические зависимости

- Подход к решению циклических динамик: решение СЛАУ (методом итераций либо методом Гаусса)
- Конь на доске: даны размеры доски, и координаты коня. Конь каждый раз ходит случайно. Требуется найти мат. ожидание числа ходов, которое он сделает, прежде чем выйдет за пределы доски
- Мышь в лабиринте: задан лабиринт со стенкам и сырами, и стартовые координаты мыши, найти мат. ожидание того, что подберёт сыр
- Настольный теннис: заданы вероятности победы каждого из игроков, и заданы результаты уже сыгранных геймов (в виде победа/поражение). Правила таковы, что если один достигает счёта 21, когда у другого счёт ≤ 19, то первый выигрывает; если же оба достигают счёта $20:20$, то счёт превращается в $15:15$
- Случайная сортировка: есть некий алгоритм, который каждый раз случайно выбирает пару индексов $i<j$ в заданном массиве $a[]$, и при этом если $a_i > a_j$, то происходит обмен. Требуется найти мат. ожидание числа обменов, которое произойдёт до момента окончания сортировки (динамика за $n \cdot n!$)
- Вышибалы: стоят $n \leq 10$ человек по кругу, для каждого задана его меткость $p[i]$ (это вероятность, с которой он попадает, когда кидает в кого-то другого). Бросают по очереди, т.е. 1-ый, 2-ой, и т.д. Игра кончается, когда остаётся один игрок. Игроки играют (т.е. выбирают цели для бросков) так, чтобы минимизировать мат. ожидание числа ходов. Требуется это самое мат. ожидание посчитать (решение — динамика по маскам d[mask][who]; заметим, что переходы по min происходят только в маски с меньшим числом битов, для которых всё можно считать уже посчитанным, а циклические переходы ведут только в нашу же маску, поэтому получаем 2^{10} циклических уровней, на каждом из которых надо решать гаусса в системе из 10 уравнений)
- Циклическая игра: есть ≤ 15 клеток, записанных по кругу, изначально стоит в первой. Ход заключается в броске кубика и переходе на заданное число клеток, при этом к своему счёту прибавляется записанное там число. Сумма всех чисел отрицательна, так что при бесконечном числе ходов мат. ожидание выигрыша — минус бесконечность. Поэтому нам разрешают в любой момент остановиться, в этом и заключается наша стратегия. Требуется найти мат. ожидание выигрыша при нашей оптимальной игре (решение — перебрать маску конечных клеток, т.е. таких клеток, при достижении которых продолжать игру точно не будем, и для каждой такой маски решить систему и посчитать мат. ожидание)

- Игра с монетами: есть ≤ 50 строк длины ≤ 10 каждая, и двое игроков. Изначально каждый выбирает по одной строке (выбирает оптимально), а затем начинается процесс случайного выбрасывания монетки до тех пор, пока в выпадаемой последовательности не встретится как подстрока какая-то из выбранных игроками строк. Нужно посчитать вероятность победы первого игрока при условии его оптимальной игры (решение — перебрать за 50^2 стартовые ходы игроков, затем произвести решение задачи, для чего заметить, что достаточно $2 \cdot 50$ состояний — это какая из двух выбранных строк текущая, и длина набранного в ней префикса)