Теория вероятностей и линейная алгебра

Задачи

Задача 1. Всего есть n тем в задачах по программированию. p_i — вероятность того, что C всеросса будет на тему i, при этом $\sum p_i = 1$. Вы решаете задачу на i-ю тему с вероятностью q_i. Какая вероятность, что вы решите C в оба дня?

Задача 2. Вы становитесь победителем всеросса с вероятностью p. Найдите вероятность того, что вы станете победителем всеросса за k участий.

Задача 3. Вы становитесь победителем всеросса с вероятностью p. Найдите матожидание количества участий во всероссе, чтобы стать победителем.

Задача 4. Дан граф из n вершин и m взвешенных рёбер. Вершины случайно и равновероятно разбивается на два множества. Найдите матожидание количества рёбер таких, что их концы лежат в разных множествах за время $\mathcal{O}(n+m)$.

Задача 5. На неориентированном графе-цикле G на 40 вершинах случайно и равновероятно выбирается раскраска вершин в черный и белый цвета. Рассмотрим подграф H, состоящий из всех вершин белого цвета и всех ребер G между вершинами белого цвета. Найдите матожидание числа компонент связности в графе H.

Задача 6. Есть 3 двери. За двумя нет ничего, за третьей — диплом победа всеросса. Вы выбрали какую-то дверь, после чего из оставшихся двух вам открыли какую-то, за которой ничего нет. Стоит ли поменять свой выбор?

— — —

Задача 7. Предложите алгоритм нахождения случайной перестановки из n элементов, работающий за время $\mathcal{O}(n)$.

Задача 8. Дан массив из n элементов в отрезке $[0; 10^{18}]$. Гарантируется, что массив был получен с помощью генератора случайных чисел. Предложите алгоритм его сортировки за ожидаемое время $\mathcal{O}(n)$.

Задача 9. Требуется в массиве найти k-ю порядковую статистику за:
а) Ожидаемое время $\mathcal{O}(n)$; **б)** Гарантированное время $\mathcal{O}(n)$.

Задача 10. Предложите способ за $\mathcal{O}(1)$ времени равновероятно выбрать случайную точку на поверхности сферы.

Задача 11. (Идеальное хеширование) Предложите способ за время $\mathcal{O}(n)$ каждому из n различных объектов сопоставить свою уникальную информацию размера $\mathcal{O}(1)$ так, чтобы потом для каждого объекта за гарантированное время $\mathcal{O}(1)$ можно было найти это число.

— — —

Задача 12. Предложите алгоритм нахождения случайной последовательности из k различных чисел от 1 до n. Порядок чисел в последовательности важен.
а) Время $\mathcal{O}(n)$; **б)** Время $\mathcal{O}(k \log n)$; **в)** Время $\mathcal{O}(k \log k)$; **г)** Время $\mathcal{O}(k)$.

Задача 13. Дан массив из n чисел. Поступают запросы: «проверить, что на отрезке $[l; r]$ есть число, встречающееся хотя бы $\frac{r-l+1}{2}$ раз». Отвечать на запрос за $\mathcal{O}(\log n)$ так, чтобы вероятность ошибки была достаточно мала.

Задача 14. Дано два списка из n чисел. Проверить, что произведения чисел в списках совпадают так, чтобы вероятность ошибки была достаточно мала. Длинную арифметику использовать нельзя. Время $\mathcal{O}(n)$

Задача 15. Даны матрицы A, B, C размера $n \times n$. Проверить, что $AB = C$ так, чтобы вероятность ошибки была достаточно мала. Время $\mathcal{O}(n^2)$.

— — —

Задача 16. Найдите матрицу перехода для рекурренты: $A_n = A_{n-1} + 2 \cdot A_{n-2} + 3 \cdot A_{n-3}$.

Задача 17. Дан граф из n вершин. Для каждого ребра в списке смежности задана вероятность перехода по нему. За один ход вы переходите в случайную вершину, смежную с текущей. Вы стоите в вершине 1.
а) Определите, с какой вероятностью вы окажетесь в вершине n за k шагов; **б)** Вы останавливаетесь, когда придёте в вершину n. Определите, с какой вероятностью вы дойдёте до n за k шагов.
Время $O(n^3 \log k)$.

Задача 18. Есть множество из n кучек, в каждой не более C камней. Изначально первый игрок может убрать некоторое неполное подмножество куч из множества. Потом второй может убрать некоторое неполное подмножество куч из оставшихся. Далее на оставшихся кучах начиная с первого оба игрока играют в НИМ. Требуется за максимально оптимальную асимптотику узнать, какое минимальное число куч мог изначально убрать первый.

Задача 19. Есть некоторая рекуррента, зависящая от последних k чисел. Вам поступают q запросов поиска i-го члена рекурренты, $i \le m$. Ответьте на запросы за время $\mathcal{O}(k^3 \log m + q \cdot k^2 \log m)$