SPb HSE, ПАДИИ, 1 курс, весна 2024/25
Практика по алгоритмам #24

День рандома
20 марта

Собрано 20 марта 2025 г. в 13:26

Содержание

День рандома

1. **Геомчик**

   Посчитать площадь объединения k сфер. Координаты центров и радиусы от 0 до 1.

2. **Понижение вероятности**

   (a) Алгоритм работает за $\mathcal{O}(n^2 \log n)$, вероятность успеха $1/\log n$.
   За какое время можно добиться вероятности успеха $1 - 1/n$?

   (b) Алгоритм работает за $\mathcal{O}(2^n)$, вероятность успеха 3^{-n}.
   За какое время можно добиться вероятности успеха $1 - 2^{-n}$?

3. **Проверка ДЗ**

   Дано ДЗ из $n{=}2k$ задач. Пусть студент в каждой из задач делает ошибку с вероятностью $\frac{1}{3}$. Ошибки во всех задача независимы.

   Алгоритм #1 проверки дз: проверить первую половину присланных решений. Пусть обнаружено 0 ошибок, какова вероятность наличия ошибок/матожидание количества ошибок?

   Алгоритм #2 проверки дз: проверить *случайную* половину присланных решений. Пусть обнаружено 0 ошибок, какова вероятность наличия ошибок/матожидание количества ошибок?

4. **2-List-Coloring**

   Придумайте жадное решение. Нужно смотреть в граф следствий.

5. **3-List-Coloring**

   Предложите рандомизированное решение за $\mathcal{O}^*(1.5^n)$.

6. **RandomShuffle**

   Дан массив, перемешайте его элементы так, чтобы все перестановки были равновероятны.

7. **Две ближайшие точки**

   Даны n точек. Используйте randomShuffle с умом.
   Научитесь за $\mathcal{O}(n)$ искать на плоскости 2 ближайшие точки. (*) А в пространстве?

8. **Простое в отрезке**

   Найдите простое число в диапазоне $[A, B]$. $B \leqslant 10^{12}$.

9. **Числа в окрестности**

   Дан массив и число ε. Известно, что $\frac{2n}{3}$ элементов массива лежат в $[x, x+\varepsilon]$ для некоторого *неизвестного* x. За линейное время выбрать $\frac{n}{3}$ элементов из $[y, y+\varepsilon]$ для некоторого y.

10. **k-й минимум**

    Дан массив. Найдите k-й минимум за $n + poly(k, \log n)$ сравнений.

11. **Матрица Татта**

Давайте запишем в матрицу смежности C неор графа вместо единиц переменные. Пусть было ребро $e = (i,j)$ тогда в клетки $C[i,j] = x_e$, а $C[j,i] = -x_e$. Что есть определитель такой матрицы (какой тип переменной)? Как проверить тождественный ли он ноль? Послушать сказку про то, как это связано с совершенными паросочетаниями.

12. **Игра на дереве**

На лекции вы изучили «дерево игры»: двое играют на бинарном дереве, первый хочет закончить игру в листе с 0, второй в листе с 1. Вы умеет решать такое за $\mathcal{O}(1.68^n)$.

Теперь усложним задачу: в листьях записаны числа, первый минимизирует, второй максимизирует. Как решить за $o(2^n)$?

13. **(*) $\alpha\beta$-отсечение**

Придумайте, как делать примерно то же, но без бинпоиска.

Придумайте, как написать бота для шахмат уровня 1-го разряда.

14. **(*) k-путь**

Пусть дан неор граф. Найдите в нём простой путь длины k (k маленькое).
Вы умеете решать это за C_n^k, нужно за $exp(k) \cdot poly(n)$.
Подсказка: а давайте покрасим вершины графа в случайные цвета от 1 до k.

Разбор задач практики

1. **Геомчик**

   Анекдот. В условии написано «площадь объединения сфер».
   Сфера – поверхность. Площадь $\cup$ сфер $= \sum_i 4\pi R_i^2$.

   Теперь всё же предположим, что мы считаем объём объединения k шариков.
   Тыкаем N случайных точек в кубе $A = [-1..2] \times [-1..2] \times [-1..2]$.
   Генерация точек: `x = -1 + 3.*rand()/RAND_MAX; y = .. ; z = ..;`

   Проверяем для точки, лежит ли она в каждой из сфер: $(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 \leqslant r_i^2$.
   Если хоть в одной лежит, `cnt++`. Ответ $= volume(A) \cdot \frac{cnt}{N} = \frac{27 \cdot cnt}{N}$. Среднее отклонение cnt – $\sqrt{N} \Rightarrow$ относительная абсолютная погрешность вычислений $\frac{27}{\sqrt{N}}$

2. **Понижение вероятности**

   a) После $\log n$ повторов вероятность ошибки $(1 - \frac{1}{\log n})^{\log n} \leqslant e^{-1}$.
   Всё это вместе $\ln n$ раз, тогда $\frac{1}{n}$. Итого $\log n \cdot \ln n$ повторов, время работы $\mathcal{O}(n^2 \log^3 n)$.

   b) $6^n \cdot n$

3. **Проверка ДЗ**

   $n{=}2k$. Если мы проверяем первую половину дз, то в каждой из оставшихся задач $\frac{1}{3}$ ошибок $\Rightarrow \mathsf{E} = \frac{1}{3} \cdot k$, вероятность наличия ошибок $1 - \left(\frac{2}{3}\right)^k$. Если мы проверяем случайную половину, ровно также. Непроверенные биты всё равно не зависят от проверенных.

4. **2-List-Coloring**

   Жадно покрасим первую вершину в любой из её двух цветов. Какие выводы мы из этого можем сделать? Если у кого-то из её соседей это был один из двух вариантов, у них остался только второй $\Rightarrow$ их тоже можно однозначно покрасить. И так далее dfs-ом. Если в процессе покраски получили противоречие (одну вершину пытаемся сразу в 2 цвета покрасить), штош, мы неверно выбрали цвет первой вершине, выберем другой. Если противоречий не нашли, прекрасно, мы покрасили часть графа, далее по индукции.

5. **3-List-Coloring**

   Если запретить случайный цвет в каждой вершине, то получится `2-List-Coloring`.
   Если $\exists$ правильная покраска, у каждой вершины мы запретили её цвет с вероятностью $\leqslant \frac{1}{3}$.
   С вероятностью $\geqslant (\frac{2}{3})^V = p$ у всех вершин не запрещён нужный цвет.

   Если повторить его $\frac{1}{p}$ раз, вероятность неудачи будет не более $(1-p)^{1/p} \approx e^{-1}$. Если повторить его $20\frac{1}{p} = \mathcal{O}(1.5^V)$ раз, вероятность неудачи будет не более $e^{-20} \approx 2 \cdot 10^{-9}$.

6. **RandomShuffle**

   Строим по индукции. Всвопываем новый элемент в случайное место перестановки.

```
    void Shuffle(int n, int *a)
        for (int i = 0; i < n; i++)
            swap(a[i], a[rand() % (i+1)]);
```

7. **Две ближайшие точки**

   Shuffle точек. Будем строить ответ по индукции. Пусть среди первых i минимум d. Плоскость разбита на клеточки $d \times d$. От номера клетки в хеш-таблице можем получить все точки в этой клетке. В каждой клетке $\leqslant$4 точек (в $\frac{d}{2}\times\frac{d}{2}$ не более 1). Пришла новая точка: перебрали точки в 9 соседних клетках. Если не нашли $<d$, и ок. Если нашли новое d, перестроили за $\Theta(i)$ всю конструкцию Время работы $\sum_i(\mathcal{O}(1)+\mathrm{Pr}_i\Theta(i))$, $\mathrm{Pr_i} = \frac{2}{i+1}$ т.к. это событие равносильно тому, что последней из $i{+}1$ мы добавили одну из тех двух, на которых достигается min расстояние.

8. **Простое в отрезке**

   Ткнём в случайное число из отрезка, проверим на простоту.

9. **Числа в окрестности**

   Простой способ.
   Ткнем в случайное число a_i, выведем в ответ все числа из $[a_i..a_i+\varepsilon]$.
   Обозначим за B минимальные $\frac{n}{3}$ чисел из $[x..x{+}\varepsilon]$. Если $a_i \in B$, то на $[a_i..a_i{+}\varepsilon]$ лежит хотя бы $\frac{n}{3}$ чисел. Вероятность попасть в B равна $\frac{n/3}{n} = \frac{1}{3}$.

   Надежный способ. Найдем статистики с номерами $\frac{n}{3}$ и $\frac{2n}{3}$, берем всё между ними.

10. **k-й минимум**

    Сделаем randomShuffle, будем поддерживать k лучших. Каждый следующий сравнивем с k-м из текущих лучших, с вероятностью $\frac{k}{i}$ сравнение скажет, да, мы должны быть среди этих k, тогда бинпоиском поймём, куда вставиться. Итого $n + k\ln n\log k$.

11. **Матрица Татта**

    $\det A = \sum_\pi(-1)^{sgn(\pi)}\prod_i a_{i,\pi_i} \Rightarrow$ в нашем случае это многочлен от нескольких переменных. Как проверить, что этот многочлен – тождественный ноль? Подставим туда случайные целые числа, посчитаем $\det A$ по простому модулю $P = 10^9{+}7$.

12. **Игра на дереве**

    Наша старая динамика: `f[v] = !f[L(v)] or !f[R(v)]`. `f[v]` – выиграет ли тот, кто ходит; он выиграет, если его соперник проиграет. Идея, если пошли в случайного и детей, и там соперник уже проиграл, то `f[v] = 1`, и от второго сына можно не запускаться.

    Новая задача: сделаем бинпоиск по ответу, получим 01-игру.

13. **(*) $\alpha\beta$-отсечение**

    Как делать без бинпоиска? Пусть мы минимизируем, а наш соперник, который максимизирует, прежде чем передать ход нам в другой ветке научился получать α, если мы наминимизируем так, что в этой ветке получится $\leqslant\alpha$, наш соперник сюда просто не пойдёт, пойдёт в α. Симметрично, тот, кто максимизирует, получил ограничение $\beta \Rightarrow f(v,\alpha,\beta)$.

    Как использовать для шахмат? Это тоже минимакс игра. Есть функция оценки, первый её минимизирует, второй максимизирует. Простейшая функция: сумма весов белых фигур минус сумма весов чёрных фигур. Можно к ней смело добавлять информацию про шахи, угрозы, связывание фигур (не может ходить, т.к. открывает шах), возможность рокировки, положение коней на краю доски и т.д.

    Как играть? Делаем перебор на глубину d. Как сделать перебор по глубже? $\alpha\beta$-отсечение отсечёт кучу веток. Как сделать ещё глубже? Будем перебирать не все, а только k самых адекватных по некоторой жадности ходов (например, запустили перебор на маленькую глубину 4, и из него сделали вывод, какие ходы получше).

14. **(*) k-путь**

    С вероятностью $\frac{k!}{k^k} \approx \frac{1}{e^k}$ наш k путь будет разноцветным. Разноцветный путь длины k можно искать динамикой за $2^k nk$. Итого нужно повторить e^k раз, получили решение за $(2e)^k nk$.

Домашнее задание

1. **(2) Площадь**

   На плоскости даны $k \leqslant 100$ прямоугольников со сторонами, параллельными осям координат и $m \leqslant 100$ кругов. Координаты вещественны. Посчитайте площадь части квадрата $[0..5] \times [0..5]$, покрытую хотя бы два раза. Принимаются только те решения, которые вы способны сесть и запрограммировать. Оцените точность полученного ответа.

2. **(2) Понижение вероятности**

   Алгоритм работает за $\mathcal{O}(n^4)$, вероятность **успеха** $1/n^2$ (ошибка односторонняя).
   За какое время можно добиться вероятности успеха $1 - \frac{1}{2^k}$ (k – параметр)?

3.1. Дополнительная часть

1. **(2) Правильный многоугольник**

   Даны $2n$ точек на плоскости. Гарантируется, что n из них образуют правильный многоугольник. Восстановить многоугольник за $\mathcal{O}(n)$.

2. **(2) Экзамен за $\mathcal{O}(1)$**

   Преподаватель хочет принять у студента экзамен. Опытный преподаватель знает: если студент плохо готов, он плавает как минимум в 50% билетов, а если студент хорошо готов, он знает не менее 80% билетов. У преподавателя есть время спросить k билетов. Всего билетов $n \geqslant k$. Придумайте алгоритм, позволяющий максимально точно различить два типа студентов.

Разбор домашнего задания

4.1. Обязательная часть

1. ?

4.2. Дополнительная часть

1. ?