Теорсеминар

Есть сложные задачи, есть простые, задачи неупорядочены по сложностям.

1. **Минимальный k-диаметр**

   Даны $n \leqslant 50$ точек на плоскости. Выбрать k точек.
   Минимизировать диаметр множества выбранных точек.

2. **Игра на массиве**

   Дан массив длины $2n \leqslant 10^5$. За ход можно откушать одно число слева или справа. Играют двое. У кого больше сумма, тот и победил. За кого из игроков есть беспроигрышная стратегия?

3. **Игра на графе**

   Дан изначально пустой граф. Двое добавляют в граф рёбра по очереди. После чьего хода появился нечётный цикл, тот редиска. Кто выиграет при оптимальной игре обоих?
   (а) $n \bmod 2 = 1$, (б) $n \bmod 4 = 0$, (в) $n \bmod 4 = 2$.

4. **Привет от Майка**

   Есть окружность. На окружности n точек. Выбрать k точек так, чтобы многоугольник на них, как на вершинах содержал центр окружности и имел максимальную площадь. $\mathcal{O}(n^2 \log n)$ или быстрее. Точно можно за $\mathcal{O}(n^2 \log k)$.

5. **Подпоследовательности**

   Дан массив, выбрать две непересекающихся возрастающих подпоследовательности максимальной (а) суммарной длины для $n \leqslant 50\,000$, (б) суммы элементов за $\mathcal{O}(n^2)$.

6. **Башня из коробок**

   Даны $n \leqslant 10^6$ коробок. У каждой есть масса m_i и прочность s_i. Выбрать максимальное число коробок так, чтобы из них можно было выстроить вертикальную башенку, в башенке на коробке i сверху можно ставить коробки суммарной массы не более s_i.
   (а) Решить. (б) Доказать корректность решения.

7. **Выполнить дерево**

   Дано $n \leqslant 10^5$ работ. У каждой есть время выполнения $t_i > 0$, и штраф $f_i > 0$. Если мы закончим выполнять i-ую работу в момент времени T_i, должны заплатить $T_i f_i$. (а) Выбрать порядок выполнения работ, минимизировать $\sum_i T_i f_i$, (б) Тоже самое, но есть зависимости, чтобы начать делать работу i, нужно сперва закончить работу p_i.

8. **Хеш множества**

   Научитесь быстро поддерживать 4 операции: добавить элемент в множество, удалить элемент из множества, `+=` к множеству (первые три операции порождают новые множества), сравнить два любых старых множества на равенство.

9. **Красно-синие точки**

   Даны $n = 500$ красных точек, 500 синих точей. Для каждого из n^3 красных треугольников насчитайте число синих точек внутри.

10. **Точки в полуплоскости**

    Даны $n \leqslant 2\,000$ точек на плоскости. Online поступают $q \leqslant 2\,000\,000$ запросов вида «сколько из данных точек лежат в полуплоскости $ax + by + c \leqslant 0$?»

11. **Расстояние Левенштейна**

    Даны две строки длины до 10^6, найти их расстояние Левенштейна, или сказать, что оно > 100.

12. **Длинные вектора**

    Дано $2n \leqslant 10^5$ векторов на плоскости. Двое играют в игру, за ход нужно забрать себе ровно один вектор, в конце сравнивают длины сумм векторов. У кого больше, тот и победит. Придумайте и реализуйте стратегию. (а) $\sum$ исходных векторов $= (0, 0)$, (б) $\forall \sum$ исходных векторов.

13. **Частное пар**

    Даны n пар $(a_i > 0, b_i > 0)$, выбрать k пар так, чтобы минимизировать $\frac{\sum a_i}{\sum b_i} \to \min$.

14. **Произведение пар**

    Даны n пар $(a_i > 0, b_i > 0)$, выбрать k пар так, чтобы минимизировать $(\sum a_i)(\sum b_i) \to \min$.

15. **Трудное Произведение пар**

    Даны n пар $(a_i > 0, b_i > 0)$, выбрать k пар так, чтобы максимизировать $(\sum a_i)(\sum b_i) \to \max$. Доказать, что задача NP-трудна.

16. **Палиндромы**

    Представить строку длины $n \leqslant 10^6$ в виде конкатенации трёх палиндромов.

Разбор задач

1. **Минимальный k-диаметр**

   Переберём пару точек A, B, которые образуют диаметр $d = |AB|$. Остальные $k-2$ точки должны лежать в пересечении P двух кругов радиуса d, AB разбивает P на две половинки $\Rightarrow$ пары точек из P, между которыми расстояние больше d образуют двудольный граф, в котором можно найти максимальное независимое множество.

2. **Игра на массиве**

   Раскрасим шахматно массив 2 цвета. Если n чётно, первый может себе гарантировать сумму любой из двух половинок. Если n нечётно, после хода первого второй может гарантировать себе сумму любой из двух половинок, но не более.

3. **Игра на графе**

   Посмотрим на последний проигрывающий ход. Перед ним граф двудольный. $a + b = n$. Если n нечётно, $a \cdot b$ обязательно чётно. Если n кратно двум, и первый, и второй могут гарантировать $a = b$, при $n \equiv 0 \bmod 4$ получаем $a \cdot b$ чётно, второй выиграет, иначе нечётно, первый выиграет.

4. **Привет от Майка**

   Первый вариант решения: dp по подотрезкам $maxArea[l, r, k]$, переход в $[l, m, \frac{k}{2}]$ и $[m, r, \frac{k}{2}]$. Состояний $\mathcal{O}(n^2 \log k)$, переходы отработают за $\mathcal{O}(n^2 \log k)$ потому что Кнут.

   Второй вариант решения. Тыкаем $\frac{n}{k}$ случайных точек, чтобы угадать какую-то из k. Для каждой пишем динамику $[n, k]$ – до куда дошли, сколько точек взяли. Переходы разделяйкой. $\frac{n}{k} \cdot nk \cdot \log n$.

5. **Подпоследовательности**

   Mincost k-flow, $k = 2$, $\mathcal{O}(n^2)$. Граф из n вершины и n^2 рёбер $\Rightarrow$ Дейкстра.

   Можно сделать $\mathcal{O}(n \log n)$ рёбер, тогда Дейкстра работает за $\mathcal{O}(n \log^2 n)$: Нам нужны рёбра $(i, a_i) \to (j, a_j)$, построим дерево mergesort и будем проводить ребра (1) в вершины этого дерево, а из неё в рёбра (2) каждый из листьев рёбер и типа (1), и типа (2) по $\mathcal{O}(n \log n)$.

   Жадное Решение для $n = 50\,000$ *и* k *последовательностей за* $\mathcal{O}(nk \log n)$. Вспомним, как мы решаем задачу за $\mathcal{O}(n \log n)$ для $k = 1$. Динамика
   $tend[\texttt{len}]$ – минимальный конец последовательности длины len.
   Теперь храним k последовательностей $\texttt{end}_i[\texttt{len}]$ для $i = 1..k$.
   Причём последовательности упорядочены по длине, end_1 – самая длинная.
   Приходит новый элемент a_i.
   Найдём бинпоиском его позицию p в $\texttt{end}_1$: $\texttt{end}_1[p-1] < a_i \leqslant \texttt{end}_1[p]$.
   Заменяем $\texttt{end}_1[p]$ на a_i. Получается, мы как бы вытолкнули из $\texttt{end}_1$ элемент $\texttt{end}_1[p]$.
   Добавим его таким же образом в $\texttt{end}_2$. И так далее...
   В итоге мы получим не последовательности, а какой-то мусор. Зато суммарная длина этого мусора в точности равна ответу на задачу. Без доказательства.

6. **Башня из коробок**

   Сортируем по $s_i + m_i$ по возрастанию (строим сверху вниз), жадно пробуем брать в таком порядке. Если не можем очередную взять, то её берём, а вместо неё выкидываем коробку максимальной массы.

   Доказательство: на каждом i-м шаге $\forall k$ верно, что префикс из k минимальных по массе коробок из выбранных – оптимальный ответ для $[i, k]$, из первых i коробок выбрать k так, чтобы получилась корректная башня минимальной массы.

7. **Выполнить дерево**

   Сортируем по частному. Чтобы обработать дерево, заметим, что как только можно выполнить оптимальную по частному работу, выгодно её сразу выполнить $\Rightarrow$ как только выполним её отца p_i, выполним и её $\Rightarrow$ смерджим их в СНМ в одну работу.

   Решение: куча по частному + СНМ.

8. **Хеш множества**

   Решение 1: $Hash(A) = \left(\sum_{a \in A} p^a\right) \bmod m$.

   Решение 2: персистентное декартово дерево от сортированного массива.

9. **Красно-синие точки**

   Прекалк для каждой из n^2 красных трапеций. Треугольник = три трапеции.

10. **Точки в полуплоскости**

    Есть n^2 различных по «порядку точек» направлений полуплоскости. Для каждого насчитаем персистентное декартово дерево, задающее порядок точек. Ответ на запрос = бинпоиск по массиву направлений + спуск по дереву. $\mathcal{O}(n^2 \log n + q \log n)$.

11. **Расстояние Левенштейна**

    Считаем только кусок матрицы динамики $[i, j]$, что $|i - j| \leqslant 100$.

12. **Длинные вектора**

    Если сумма ноль, всегда ничья. Если сумма не ноль, то пусть она равна v, тогда перпендикулярные v направления у игроков в конце игры равны, а коллинеарную v часть они и будут максимизировать $\Rightarrow$ выбираем вектор с максимальным скалярным произведением с v.

13. **Частное пар**

    Бинпоиск по ответу. $\frac{\sum a_i}{\sum b_i} \geqslant x \Leftrightarrow \sum (a_i - x b_i) \geqslant 0$, выберем k максимальных за линию.

14. **Произведение пар**

    Переберём направление суммы (n^2 различных штук), для каждого выберем k наименьших проекций. Один из n^2 вариантов окажется оптимальным.

15. **Трудное Произведение пар**

    Решает subsetsum при $a_i + b_i = C$ и $k = \frac{n}{2}$.

16. **Палиндромы**

    Утверждение: выгодно жадно взять или самый длинный слева, или самый длинный справа.