SPb HSE, 1 курс, осень 2022/23
Практика по алгоритмам #11

Динамика, комбинаторика

7 декабря

Собрано 9 декабря 2022 г. в 10:03

Содержание

Динамика, комбинаторика

1. **Самое дешёвое поддерево размера k**

   Дано дерево. Веса в вершинах.
   Выбрать связный подграф дерева, содержащий корень, из ровно k вершин min веса.

   a) Двоичное дерево. $n, k \leqslant 100$.
   b) Любое дерево. $n, k \leqslant 100$.
   c) (*) А как сделать за $\mathcal{O}(n^2)$?
   d) (**) А как сделать за $\mathcal{O}(nk)$?

2. **k-е разбиение**

   Найдите k-е лексикографически разбиение числа n на убывающие слагаемые. $\mathcal{O}(n^2)$.
   Разбиение – вектор $a_1 > a_2 > \ldots > a_m$.

3. **LR-задача**

   Посчитать количество чисел, состоящих из цифр $a_1, a_2, \ldots, a_d$. $d \leqslant 10$.

   a) кратных M из ровно k цифр, за $\mathcal{O}(kM)$.
   b) на отрезке $[L, R]$ за $\mathcal{O}(k)$, $1 \leqslant L \leqslant R < 10^k$.
   c) (*) и кратных M и на отрезке за $\mathcal{O}(kM)$.

4. **(*) Битоническая задача о коммивояжера**

   Найдите во взвешенном графе гамильтонов цикл минимального веса, который удовлетворяет дополнительно следующему свойству: сначала номера посещенных вершин возрастают, а затем убывают. Время $\mathcal{O}(n^2)$.

5. **Различные и далёкие**

   Дан массив объектов. У каждого объекта есть стоимость $cost_i$ и тип $type_i \leqslant k$. Выберите множество объектов максимальной суммарной стоимости так, что у каждой пары объектов типа различны, а разность индексов хотя бы d. $n \leqslant 100$, $k \leqslant 10$.

6. **Перебор всего подряд**

   За сколько работает следующий код?

```
for (множество A):
    for (B ⊆ A):
        for (C ⊇ A):
            makeOneMoreHumanHappy()
```

7. **Надмножества**

   Придумайте цикл, который перебирает надмножества.

8. **(*) Возрастающие подмножества**

   Умеем писать цикл, который по маске A перебирает маски $B \subseteq A$. Он делал это в убывающем порядке. Придумайте цикл, который перебирает их в возрастающем порядке.

9. **Кратчайший цикл**

   Даны k выделеных вершин в графе из n и расстояния между всеми парами вершин. Найдите кратчайший цикл в графе, проходящий через все выделенные вершины. В этой задаче по каждой вершине можно ходить несколько раз. $\mathcal{O}(2^k n^2)$, (*) $\mathcal{O}(2^k n)$.

10. **Ровное распределение**

    Даны n вещей с весами a_i. Разбить их на k кучек так, чтобы суммарный вес каждой кучки был ровно s_j. $2 \leqslant k \leqslant n \leqslant 18$, $1 \leqslant a_i, s_j \leqslant 10^{18}$.

11. **Перевозка**

    Есть k грузовиков, у каждого есть своя вместимость W_i. Есть n вещей, у каждой есть вес a_j. Один заезд – погрузить и отправить все грузовики.
    Перевезти все вещи минимальным числом заездов.

    a) $k = 1$, $\mathcal{O}(3^n)$
    b) $k = 2$, $\mathcal{O}(4^n)$, $\mathcal{O}(3^n)$
    c) $\mathcal{O}(3^n k)$
    d) $\mathcal{O}(2^n n)$

12. **(*) Покрытие строки**

    Покрыть строку s минимальным числом строк $t_i \in T$. Каждую из строк t_i мы или берём (тратим 1), или не берём (тратим 0). Если взяли, можно использовать несколько раз, как подстроку s. Нужно, чтобы каждый символ s был покрыт. $\mathcal{O}^*(2^{\min(|s|,|T|)})$.

13. **(*) Математика или динамика?**

    a) Сколько существует правильных скобочных последовательностей из $2n$ скобок?
    b) Сколько существует почти правильных скобочных последовательностей из n скобок (в конце осталось ровно a не закрытых скобок).
    c) А если a не фиксировано (любое не закрытое число скобок)?
    d) Сколько способов n одинаковых объектов разбить на упорядоченный набор из k множеств (две версии: множества могут быть пустыми и нет)?
    e) Сколько способов n разных объектов разбить на набор из k множеств одинакового размера (две версии: набор упорядочен и нет)?
    f) Сколько способов n разных объектов разбить на набор из k множеств произвольного размера (две версии: набор упорядочен и нет)?
    g) Сколько существует мультимножеств из n элементов, каждый элемент от 1 до k?
    h) Известно, что есть k различных деревьев из v вершин с точностью до изоморфизма. Сколько существует различных множеств из n деревьев из v вершин каждое?

Разбор задач практики

1. **Самое дешёвое поддерево размера k**

   Дано дерево $\Rightarrow$ динамика по дереву $\Rightarrow$
   нужно подвесить дерево за любую вершину и $\forall v$ что-то хранить для поддерева v.

   Динамика: $f[v,i]$ – минимальный вес обрубка поддерева v размера ровно i (саму v обязательно берём). Если сможем такое насчитать, $ans = \min_v f[v,k]$.

   Пересчёт: запустим рюкзак по детям v: $g[v,j,i]$ — рассмотрели первых j детей, набрали в рюкзак ровно i вершин $\Rightarrow$ $f[v,0]=0, f[v,i]=g[v,deg_v,i]$, $g[v,0,1]=weight_v$.
   Добавить одного ребёнка = перебрать сколько x вершин мы берём у него:
   $g[v,j{+}1,i] = \min_x(g[v,j,i{-}x] + f[child[v,j],x])$.

   Как же сделать за $\mathcal{O}(nk)$? А оно уже за столько работает, если правильно написать. Давайте делать динамику вперёд и, когда добавляем j-го ребёнка перебирать сколько вершин берём слева, сколько справа за $\min(\text{размера поддерева}, k)$.

   Версия для бинарного дерева:

   Переход: пусть у вершины v всего два сына – a, b. Уже посчитаны $f[a]$, $f[b]$, посчитаем $f[v]$ за $\mathcal{O}(\min(size_a,k)\cdot\min(size_b,k))$. $\min(size_a,k)$ – кол-во ненулевых элементов в $f[a]$. Осталось доказать время работы...

   Докажем n^2 по индукции. Пусть $size_v = n$, $size_a = x$, $size_b = n-x-1$, тогда
   $T(n) \leqslant \max_x\big[T(x)+T(n-x-1)+x(n-x-1)\big] \leqslant (n-1)^2 \leqslant n^2$.
   Доказательство $\mathcal{O}(nk)$ можно прочесть здесь [SK].

2. **k-е разбиение**

   Следуем общему алгоритму с лекции: строим разбиение слева направо, дописываем по одному числу. Нужно уметь отвечать на вопрос «сколько способов закончить»?

   $p[n,m]$ – число разбиений n на различные слагаемые не более m.
   Либо есть слагаемое, равное m, либо нет: $p[n,m] = p[n{-}m, m{-}1] + p[n, m{-}1]$.

   Если уже выписали i слагаемых $a_1 > a_2 > \dots > a_i \Rightarrow p[n{-}\sum a_i, a_i{-}1]$ способов закончить.

3. **LR-задача**

   Кратных M из ровно k цифр, за $\mathcal{O}(kM)$.
   Динамика $f[k,rest]$ – число способов выписать k цифр, получить остаток $rest$.
   Пересчёт вперёд: `for k: for rest: for d: f[k+1,(rest*BASE+d)%M] += f[k,rest]`.

   На отрезке $[L,R]$ за $\mathcal{O}(k)$, $1 \leqslant L \leqslant R < 10^k$.
   Сперва разобьём отрезок на префиксы: $\mathtt{ans}[L,R) = \mathtt{ans}[0,R) - \mathtt{ans}[0,L)$.

   Пусть всего t различных цифр, тогда всего t^k чисел длины k.

   Учтём числа длины меньше $|R|$: $\sum_{i=0..|R|-1} t^i$. Числа длины ровно $|R|$, меньшие чем $|R|$ имеют вид «сколько то j цифр совпало с R, $(j{+}1)$-я цифра x меньше».
   Переберём j, переберём x. Есть $t^{|R|-j-1}$ способов закончить число. Степени t предподсчитаны.

   И кратных M, и на отрезке за $\mathcal{O}(kM)$. Меняем выше все t^k на значение динамики $f[k,rest]$.

4. **Битоническая задача о коммивояжере**

   Переформулируем: у нас есть два возрастающих пути, приходящих в вершину n.

   Состояние: уже просмотрели вершины $1, 2, \ldots, k$, первый путь кончается на вершину i, второй на j. $f[i,j]$ – минимальная цена таких путей.

   Динамика вперед: $k = \max(i,j)+1$, $\texttt{relax}(f[k,j], f[i,j]+w[i,k])$, $\texttt{relax}(f[i,k], f[i,j]+w[j,k])$.

5. **Различные и далёкие**

   Делаем динамику из перебора. Идём по массиву слева направо, каждый элемент массива или берём, или не берём. По ходу перебора важно помнить, какие типы уже использовали (маска $mask$ из k бит), какой элемент $last$ взят последним, сколько i уже рассмотрели.

   Итого: $cost[mask,last,i] \to \max$, из каждого состояния 2 перехода (взять, не взять). $\mathcal{O}(2^k{\cdot}n{\cdot}n)$.

   Можно хранить $cost[mask,last] = \max_{j \leqslant last-d} cost[mask-2^{last}, j]$, переход за $\mathcal{O}(1)$ префиксным минимумом. Итого $\mathcal{O}(2^k \cdot n)$.

   Можно хранить $cost[mask,i] = \max(cost[mask,i-1], cost[mask-2^{a_i}, i-d])$. Первый переход «не берём i», второй — «берём i». Второй переход делаем, если 2^{a_i} есть в $mask$. $\mathcal{O}(2^k \cdot n)$.

6. **Перебор всего подряд**

   $B \subseteq A \subseteq C \Rightarrow$ у каждого элемента 4 варианта, где быть — $B, A \setminus B, B \setminus C, \overline{C} \Rightarrow \Theta(4^n)$.

7. **Надмножества**

   ```
   for (B = A; B < (1 << n); B = (B + 1) | A)
   ```

8. **Возрастающие подмножества**

   Перебираем $B \subseteq A$. Обозначим за i позицию младшего бита $D = A \setminus B$.
   Нужно сделать `B |= 2`^{i}, затем занулить хвост: `B &= ~(2`^{i}`-1)`.
   Осталось заметить, что `2`^{i} `= D ^ (D & (D-1))`

   Другое решение: перебираем по убыванию, но по ходу получаем и обратный порядок тоже
   `for (B = A; B > 0; B--, B &= A) B' = A - B; // B'` по возрастанию

9. **Кратчайший цикл**

   $minCost[mask, v]$ — уже посетили $mask$ интересных, сейчас стоим в v (необязательно интересная). Из каждого состояния n переходов — перебрать, куда пойдём из v. $\mathcal{O}(2^k n^2)$.
   Начальное состояние: посетили первую из особых a_1, стоим в ней.
   Конечное: выбираем min по всем $minCost[2^k-1, v]$, что из v есть ребро в a_1.

   Чтобы получить $\mathcal{O}(2^k n)$, воспользуемся такой же битовой магией, как на лекции.

10. **Ровное распределение**

    $is[k, mask]$ — можно ли распределить множество $mask$ на k первых рюкзаков.

    Переход для динамики назад: перебираем подмножества
    $m \subseteq mask\colon sum[m] = s_k$, где $sum[m]$ — сумма весов в маске m. $\mathcal{O}(3^n k)$.

11. **Перевозка**

    Прежде всего предпосчитаем вес каждого подмножества $\texttt{sum}[T]$ за $\mathcal{O}(2^n)$.

a) $k = 1$, $\mathcal{O}(3^n)$. Для всех S и всех $T \subseteq S$, если $\texttt{sum}[T] \leqslant W_1$, $\texttt{relax}(d[S], d[S \setminus T] + 1)$.

b) $k = 2$, $\mathcal{O}(4^n)$. Для всех S перебрать $T \subseteq S$ – что перевезено последним заездом.
Для T перебрать $T_1 \subseteq T$ – что везёт первый грузовик.
Если $\texttt{sum}[T_1] \leqslant W_1$ и $\texttt{sum}[T \setminus T_1] \leqslant W_2$), $\texttt{relax}(d[S], d[S \setminus T] + 1)$.

Можно для каждого T за $\mathcal{O}(3^n)$ предпосчитать, вместится ли оно в два грузовика.
`for S: for T: for T₁: if relax` $\to$ `for T: for T₁: if` $+$ `for S: for T: if relax`

c) $\mathcal{O}(3^n k)$. Сначала для каждого T посчитаем, $\texttt{can}[T]$ можно ли его увезти за один заезд.
Снова почти рюкзак. $\texttt{f}[T, i]$ – можно ли увезти T первыми i машинами, $\texttt{can}[T] = \texttt{f}[T, k]$.
`f[T,i] |= f[T \ T₁, i-1] && sum[T₁] <= W[i]`.
Далее минимизируем число заездов, как в прошлом пункте.

d) $\mathcal{O}(2^n n)$. Измельчение перехода! Действие – погрузить ещё один предмет. Грузовики заполняем по очереди. Для каждого S минимизируем тройку $\langle runs, trucks, last\rangle$.

`runs` – сколько уже сделано заездов,
`trucks` – сколько грузовиков уже погружено,
`last` – какой вес в еще не догруженном грузовике.

Переход: взять предмет i, которого нет в S, и

- погрузить в последний грузовик: $\langle r, t, l\rangle[S] \to \langle r, t, l{+}a[i]\rangle[S \cup \{i\}]$,
- начать новый грузовик: $\langle r, t, l\rangle[S] \to \langle r, t{+}1, a[i]\rangle[S \cup \{i\}]$,
- сделать новый заезд: $\langle r, t, l\rangle[S] \to \langle r{+}1, 0, a[i]\rangle[S \cup \{i\}]$.

P.S. Достаточно хранить $\langle r \cdot k + t, l\rangle$.

12. **(*) Покрытие строки**

Пусть $|s| = m$, $|T| = n$, w – размер машинного слова.
Решение $\mathcal{O}(mn + 2^n \frac{m}{w})$ – для каждого слова из T предподсчитали маску позиций в s, которые покроет слово. Рекурсивно перебираем, какие из слов T мы берём, при спуске в рекурсию поддерживаем объединение масок.

За $\mathcal{O}^*(2^{|s|})$. Динамика $\texttt{opt}[A, i]$ – минимальное число строк среди $T[1..i]$, которыми можно покрыть множество A позиций строки s. Опять рюкзак =) Асимптотика $2^m n$.

13. **(*) Математика или динамика?**

a) ПСП. $\binom{2n}{n}/(n+1) = \binom{2n}{n} - \binom{2n}{n-1}$ (число Каталана).

b) Почти ПСП. $\binom{2n}{n+a/2} - \binom{2n}{n+a/2-1}$ (придумывается, как формула для Каталана).

c) Почти ПСП. Положительных, но, возможно, уходящих в минус ровно $\frac{1}{2}(2^{2n} - \binom{2n}{n})$.
Далее отразим, как и в предыдущих. Опять разность цешек.

d) n одинаковых объектов на k множеств.
Непустых: $\binom{n-1}{k-1}$ – ставим $k-1$ перегородку на $n-1$ позицию.
С пустыми: перегородки могут повторяться, вариантов перегородок теперь $n+1$,
ответ $= \binom{n+k-1}{n} = \binom{n+k-1}{k-1}$ (сочетание с повторениями).

e) На k множеств равного размера. Упорядочены: $\frac{n!}{(\frac{n}{k}!)^k}$. Не упорядочены: $\frac{n!}{(\frac{n}{k}!)^k k!}$.

f) На k множеств произвольного размера. Упорядочены: k^n.
Не упорядочены: динамика `count[n,k] = count[n-1,k-1] + k * count[n-1,k]`

(либо `n` в отдельном множестве, либо в одном из `k`).

g) Мультимножества из n элементов от 1 до k – ровно пункт (c).

h) Различных множеств из n деревьев из v вершин – ровно (f). Пригодится в доп-дз.

Домашнее задание

3.1. Обязательная часть

1. **(2) k-е разбиение в другом порядке**

   Найдите k-е лексикографически разбиение числа n на возрастающие слагаемые. $\mathcal{O}(n^2)$.
   Разбиение – вектор $a_1 < a_2 < \dots < a_m$

2. **(2) Странные числа**

   Даны B, R. Посчитайте количество таких чисел x, что

   a) $x \leqslant B$
   b) В записи числа используются только нечётные цифры.
   c) Если прочитать число задом наперёд, то $\overleftarrow{x} \leqslant R$.

   Ограничения: $1 \leqslant B, R \leqslant 10^{18}$.

3. **(2) k-е странное число**

   Найдите k-е по величине число из описанных в предыдущей задаче. Можно пользоваться предыдущей как решенной.

4. **(2) Подмножества надмножеств подмножеств**

   Оцените сложность кода. Ответ следует обосновать.

```
for (a = 0; a < (1 << n); ++a)
    for (b = a; b < (1 << n); b = (b + 1) | a)
        for (c = b; c > 0; c = (c - 1) & b)
            ;
```

5. **(2) Ещё более странные числа**

   Даны k, L, R, A, B.
   Найдите k-е по величине число среди таких, что выполнены все два свойства:

   a) $A \leqslant x \leqslant B$
   b) Если прочитать число задом наперёд, то $L \leqslant \overleftarrow{x} \leqslant R$.

   Ограничения: $1 \leqslant k, A, B, L, R \leqslant 10^{18}$.
   Указание: сведите задачу к предыдущей, можно пользоваться предыдущей, как решенной.

6. **(3) Циклы в неориентированном графе**

   Дан граф. Для каждого подмножества вершин A проверить, есть ли простой цикл, проходящий по всем вершинам A ровно один раз (и только по вершинам из A).

   a) **(2)** $\mathcal{O}(2^n n^2)$.
   b) **(3)** $\mathcal{O}(2^n n)$.

7. **(2) Рюкзак в несколько заходов**

   Есть n вещей, у каждой есть стоимость v_i и вес w_i. Есть рюкзак, в котором можно унести набор вещей суммарного веса не более W за один подход. За $m = 2^k$ подходов унести вещи максимальной суммарной стоимости. Время $\mathcal{O}(3^n k)$.
   (+1) балл, если работает не только для m – степеней двойки.

8. **(2) Set Cover**

   Даны $A, B_1, \dots, B_m \subseteq \{0, \dots, n-1\}$. Выбрать минимальный набор $\{B_{i_j}\}$: $\bigcup B_{i_j} = A$.
   Чтобы везде можно было бесплатно применять битовую магию, предположим $n, m \leqslant 64$.

   a) **(1)** $\mathcal{O}(2^m)$. И ни на n больше!
   b) **(1)** $\mathcal{O}(2^n m)$.

3.2. Дополнительная часть

1. **(3) Перестановки**

   Сколько существует перестановок, у которых все длины циклов хотя бы k?

   a) **(1.5)** $\mathcal{O}(n^2)$
   b) **(3)** $\mathcal{O}(n)$

2. **(2) НОПП**

   Для двух последовательностей длины n за $\mathcal{O}(n^2)$ найти наибольшую общую пилообразную подпоследовательность.

3. **(4) Свертка**

   $(f * g)[A] = \sum_{B \cup C = A} f[B] \cdot g[C]$. Даны f и g, посчитать $f * g$.

4. **(5) Сколько деревьев?**

   Посчитайте число различных деревьев из n вершин с выделенным корнем. Изоморфные деревья считаются одинаковыми, при изоморфизме корень должен переходить в корень.
   Например, из 2 вершин есть 1 дерево, из 3 вершин 2 дерева, а из 4 вершин 4 дерева.

   a) **(1.5)** $\mathcal{O}(\text{poly}(n))$.
   b) **(3)** $\mathcal{O}(n^4 \text{poly}(\log n))$.
   c) **(5)** $\mathcal{O}(n^3 \text{poly}(\log n))$.

   Существует решение за $\mathcal{O}(n^2 \log n)$.