

SPb HSE, 1 курс, осень 2024/25

Практика по алгоритмам #11

Динамика 3: пробуждение силы

27 ноября

Собрано 1 декабря 2024 г. в 19:37

---

## Содержание

1. Динамика 3: пробуждение силы	1
2. Разбор задач практики	3
3. Домашнее задание	6
3.1. Обязательная часть . . . . .	6
3.2. Дополнительная часть . . . . .	7

## Динамика 3: пробуждение силы

### 1. Самое дешёвое поддерево размера $k$

Дано дерево. Веса в вершинах. Выбрать связное поддерево из  $k$  вершин  $\min$  веса.

(\*) А как сделать за  $\mathcal{O}(n^2)$ ?

(\*) 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  $a_1 > a_2 > \dots > a_m$ .

### 3. LR-задача

Посчитать количество чисел, состоящих из цифр  $a_1, a_2, \dots, a_d$ .

a) кратных  $M$  из ровно  $k$  цифр, за  $\mathcal{O}(kM)$ .

b) на отрезке  $[L, R]$  за  $\mathcal{O}(k)$ ,  $1 \leq L \leq R < 10^k$ .

c) и кратных  $M$  и на отрезке за  $\mathcal{O}(kM)$ .

### 4. Битоническая задача о коммивояжере

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  $type_i \leq k$ . Выберите множество объектов максимальной суммарной стоимости так, что у каждой пары объектов типы различны, а разность индексов хотя бы  $d$ .  $n \leq 100$ ,  $k \leq 10$ .

### 6. Надмножества

Придумайте цикл, который перебирает надмножества.

### 7. 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.

### 8. 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)  $\mathcal{O}(3^nk)$

d)  $\mathcal{O}(2^nn)$

**9. 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  $n, m \leq 64$ .

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

**10. (\*) 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.  $\mathcal{O}^*(2^{\min(|s|, |T|)})$ .

**11. (\*) Математика или динамика?**

- a) Сколько существует правильных скобочных последовательностей из  $2n$  скобок?
- b) Сколько существует почти правильных скобочных последовательностей из  $n$  скобок (несколько скобок в конце остались открытыми)?
- c) Сколько способов  $n$  одинаковых объектов разбить на упорядоченный набор из  $k$  множеств (две версии: множества могут быть пустыми и нет)?
- d) Сколько способов  $n$  разных объектов разбить на набор из  $k$  множеств одинакового размера (две версии: набор упорядочен и нет)?
- e) Сколько способов  $n$  разных объектов разбить на набор из  $k$  множеств произвольного размера (две версии: набор упорядочен и нет)?
- f) Сколько существует мультимножеств из  $n$  элементов, каждый элемент от 1 до  $k$ ?
- g) 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

$$T(n) \leq \max_x [T(x) + T(n - x - 1) + x(n - x - 1)] \leq (n - 1)^2 \leq n^2. \text{ Доказательство } \mathcal{O}(nk)$$

можно прочесть здесь: <http://acm.spbgu.ru/~sk1/download/codeforces/NK.pdf>

### 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  $a_1 > a_2 > \dots > a_i$ , то есть  $p[n - \sum a_i, a_i - 1]$  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:  $f[k+1, (rest+d)\%M] += f[k, rest]$ .

*На отрезке  $[L, R]$  за  $\mathcal{O}(k)$ ,  $1 \leq L \leq R < 10^k$ .*

Сперва разобьём отрезок на префиксы:  $ans[L, R] = ans[0, R] - ans[0, L]$ .

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  $1, 2, \dots, k$ , первый путь кончается на вершину  $i$ , второй на  $j$ .  $f[i, j]$  – минимальная цена таких путей.

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

## 5. Различные и далёкие

### 6. Надмножества

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

### 7. Возрастающие подмножества

Перебираем  $B \subseteq A$ . Обозначим за  $i$  позицию младшего бита  $D = A \setminus B$ .

Нужно сделать  $B |= 2^i$ , затем занулить хвост:  $B \&= \sim(2^i - 1)$ .

Осталось заметить, что  $2^i = D \wedge (D \& (D - 1))$

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`

## 8. Перевозка

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

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

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  $\text{sum}[T_1] \leq W_1$  и  $\text{sum}[T \setminus T_1] \leq W_2$ ,  $\text{relax}(d[S], d[S \setminus T] + 1)$ .

Можно для каждого  $T$  за  $\mathcal{O}(3^n)$  предсчитать, вместится ли оно в два грузовика.

`for S: for T: for T1: if relax → for T: for T1: if + for S: for T: if relax`

c)  $\mathcal{O}(3^nk)$ . Сначала для каждого  $T$  посчитаем,  $\text{can}[T]$  можно ли его увезти за один заезд.

Снова почти рюкзак.  $f[T, i]$  – можно ли увезти  $T$  первыми  $i$  машинами,  $\text{can}[T] = f[T, k]$ .

`f[T, i] |= f[T \setminus T1, i-1] \&\& sum[T1] <= W[i]`.

Далее минимизируем число заездов, как в прошлом пункте.

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

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

p.s. Достаточно хранить  $\langle r \cdot k + t, l \rangle$ .

## 9. Set Cover

$\mathcal{O}(2^m)$ . Рекурсивный перебор, который битовым OR считает объединение.

$\mathcal{O}(2^nm)$ . Динамика  $f[X]$  – минимальное число  $B$ -шек, нужное, чтобы набрать  $X$ .

Переходы для динамики вперед: `for X for i relax(f[X | Bi], f[X] + 1)`.

10. (\*) **Покрывание 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 покрое слово. Рекурсивно перебираем, какие из слов  $T$  мы берём, при спуске в рекурсию поддерживаем объединение масок.

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

11. (\*) **Математика или динамика?**

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

b) Почти ПСП. Разность двух цэшек (придумывается, как формула для Каталана)

c)  $n$  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).

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

e) На  $k$  множеств произвольного размера. Упорядочены:  $k^n$ .

Не упорядочены: динамика  $\text{count}[n, k] = \text{count}[n-1, k-1] + k * \text{count}[n-1, k]$   
(либо  $n$  в отдельном множестве, либо в одном из  $k$ ).

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

g) 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 чисел  $x$ , для которых выполняются 3 условия:

- $x \leq B$
- В записи числа используются только нечётные цифры.
- Если прочитать число задом наперёд, то  $\overleftarrow{x} \leq R$ .

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

### 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.

```

1 for (a = 0; a < (1 << n); ++a)
2     for (b = a; b < (1 << n); b = (b + 1) | a)
3         for (c = b; c > 0; c = (c - 1) & b)
4             ;

```

### 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:

- $A \leq x \leq B$
- Если прочитать число задом наперёд, то  $L \leq \overleftarrow{x} \leq R$ .

Ограничения:  $1 \leq k, A, B, L, R \leq 10^{18}$ .

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  $A$ ).

- (2)  $\mathcal{O}(2^n n^2)$ .
- (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  $\mathcal{O}(3^k)$ .

(+1) балл, если работает не только для  $m$  – степеней двойки.

## 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) НОП

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