Первый курс, осенний семестр 2018/19
Практика по алгоритмам #10

Динамика, комбинаторика

10 ноября

Собрано 10 ноября 2018 г. в 23:27

Содержание

Динамика, комбинаторика

1. **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?
2. **k-е разбиение**
   Найдите k-е лексикографически разбиение числа n на убывающие слагаемые. $\mathcal{O}(n^2)$.
   Разбиение – вектор $a_1 > a_2 > \ldots > a_k$.
3. **LR-задача**
   Посчитать количество чисел кратных M на отрезке $[L, R]$, состоящих из определенных цифр. $1 \leqslant L \leqslant R < 10^k$. $\mathcal{O}(kM)$.
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. **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.
6. **Надмножества**
   А теперь придумайте цикл, который перебирает надмножества.
7. **Перевозка**
   Есть k грузовиков с заданной вместимостью, перевезти n вещей с заданными весами минимальным числом заездов. Один заезд – погрузить и отправить все грузовики.
   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) $\mathcal{O}(2^n n)$
8. **(*) 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. $\mathcal{O}^*(2^{\min(|s|,|T|)})$.

Разбор задач практики

1. **Математика или динамика?**
   a) ПСП. $\binom{2n}{n}/(n+1) = \binom{2n}{n} - \binom{2n}{n-1}$ (число Каталана).

   b) Почти ПСП. $2^n - \sum\limits_{k=0}^{\lfloor\frac{n-1}{2}\rfloor} \binom{n}{k}$.

   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 `count[n][k] = count[n - 1][k - 1] + k * count[n - 1][k]` (либо `n` в отдельном множестве, либо в одном из `k`).

   f) Мультимножества из n элементов от 1 до k – снова пункт c!

   g) Различных множеств из n деревьев из v вершин – предыдущий пункт.

2. **k-е разбиение**
   $p[n, m]$ – число разбиений n на различные слагаемые не более m. Либо есть слагаемое m, либо нет: $p[n, m] = p[n-m, m-1] + p[n, m-1]$.
   Разбиения с первым слагаемым n имеют номера $\big[0, p[n-n, n-1]\big)$. С первым слагаемым $n-1$ разбиений $p[n-(n-1), n-2]$, их номера $\big[p[n-n, n-1], p[n-n, n-1] + p[n-(n-1), n-2]\big)$.
   И так далее, с первым слагаемым s – $\big[\sum\limits_{m>s} p[n-m, m-1], \sum\limits_{m\geqslant s} p[n-m, m-1]\big)$.
   По k нашли первое слагаемое s, таких разбиений $p[n-s, s-1]$, рекурсивно запускаемся от $n-s, k-p[n-(s+1), s]$.

3. **LR-задача**
   $\texttt{ans}[L, R] = \texttt{ans}[0, R] - \texttt{ans}[0, L-1]$.
   Переберем, сколько цифр в нашем числе совпадает с R. Если первые i совпадают, то $(i+1)$-я цифра меньше, переберем и ее. Остается узнать, сколько бывает чисел длины $k-i-1$ с заданным остатком при делении на M.
   Динамика $f[k, r]$ – сколько чисел длины k с остатком r. Для всех s делаем $f[k+1, (10r+s) \bmod M]$ `+=` $f[k, r]$.

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

5. **Возрастающие подмножества**
   Перебираем $B \subseteq A$.

Берем ближайший к краю 0, который можно сделать 1. Это младший бит $A \setminus B$, он равен $x = 2^i$.
Нужно сделать B `|=` x, затем занулить хвост: B `&=` $(x - 1)$.

6. **Надмножества**
`for (B = A; B < (1 << n); B = (B + 1) | A)`

7. **Перевозка**
Прежде всего предпосчитаем вес каждого подмножества за $\mathcal{O}(2^n)$.
a) $k = 1$, $\mathcal{O}(3^n)$ – для всех S и всех $T \subseteq S$ $\texttt{relax}(d[S], d[S \setminus T] + 1)$ при $\texttt{sum}[T] \leqslant W$.
b) $k = 2$, $\mathcal{O}(4^n)$ – для всех S перебрать $T \subseteq S$, которое перевезено последним, для T перебрать, какое $T_1 \subseteq T$ идет в первый грузовик. $\texttt{relax}(d[S], d[S \setminus T] + 1)$ при $\texttt{sum}[T_1] \leqslant W_1$ и $\texttt{sum}[T \setminus T_1] \leqslant W_2$).
Но можно для каждого множества за $\mathcal{O}(3^n)$ предпосчитать, вместится ли оно в два грузовика.
Затем за $\mathcal{O}(3^n)$ минимизировать число заездов: для всех S и всех $T \subseteq S$
`if (can[T]) relax(d[S], d[S \ T] + 1).`
c) $\mathcal{O}(3^n k)$. Сначала для каждого множества считаем, можно ли его увезти за один заезд, снова почти рюкзак: для всех S и всех $T \subseteq S$
`can[S, i] |= can[S \ T, i-1] && sum[T] <= W[i].`
Минимизируем число заездов, как в прошлом пункте.
d) $\mathcal{O}(2^n n)$. Для каждого S минимизируем тройку $\langle runs, trucks, last \rangle$ – сколько уже сделано заездов, сколько грузовиков уже погружено, какой вес в еще не догруженном грузовике.
Переходы: взять предмет i, которого нет в S, и:
- погрузить в последний грузовик: $\langle r, t, l \rangle[S] \to \langle r, t, l + w[i] \rangle[S \cup \{i\}]$,
- начать новый грузовик: $\langle r, t, l \rangle[S] \to \langle r, t + 1, w[i] \rangle[S \cup \{i\}]$,
- сделать новый заезд: $\langle r, t, l \rangle[S] \to \langle r + 1, 0, w[i] \rangle[S \cup \{i\}]$.

8. **(*) 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 $2^m n$.

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 $a_1 < a_2 < \dots < a_k$

2. **(3) Странные числа**
   Даны k, L, R, A, B.
   Найдите k-е по величине число среди таких, что выполнены все три свойства:
   a) $A \leqslant x \leqslant B$
   b) В записи числа используются только нечётные цифры
   c) Если прочитать число задом наперёд, то $L \leqslant \overleftarrow{x} \leqslant R$.

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

3. **(1) Подмножества надмножеств подмножеств**
   Оцените сложность кода:

```
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)
      ;
```

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

5. **(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.

6. **(3) Set Cover**
   Даны $A, B_1, \dots, B_m \subseteq \{0, \dots, n-1\}$.
   Выбрать минимальный набор $\{B_{i_j}\}$: $\bigcup B_{i_j} = A$.
   a) **(1)** $\mathcal{O}(2^m)$.
   b) **(2)** $\mathcal{O}(2^n m)$.

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. **(3) Разбиения на различные слагаемые**
   Найти число разбиений n на различные неупорядоченные слагаемые. $\mathcal{O}(n\sqrt{n})$.

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