

Московский физико-технический институт
Физтех-школа прикладной математики и информатики

БОЛЬШОЕ НАЗВАНИЕ КУРСА
V СЕМЕСТР

Лектор: *Иван Иванович Иванов*

h/ν

Автор: *Павел Дуров*
Репозиторий на Github

осень 2022

Содержание

1	Динамическое программирование (окончание)	2
1.1	ДП с помощью масок	2
1.1.1	Задача 1	2
1.1.2	Задача 2	2
1.1.3	Задача 3	3
1.2	ДП по профилю	3
1.2.1	Задача 1	3
1.3	Изломанный профиль	3

1 Динамическое программирование (окончание)

1.1 ДП с помощью масок

Пусть $U = \{0, 1, \dots, n-1\}$ $A \subset U$

Тогда будем записывать A как массив длины n , где $i \in A \leftrightarrow a_i = 1$. Такое представление называется *маской множества*

Как проверить входит ли x в A ?

```
bit(mask, pos) {
    return (mask » pos) & 1; }
```

Как брать пересечения и объединения?

$A \cap B$	$A \cup B$
$mask_A \& mask_B$	$mask_A \vee mask_B$

1.1.1 Задача 1

Пусть даны a_{ij} - стоимость выполнения j -го задания i -ым работником. Найти минимальную стоимость выполнения всех заданий.

РЕШЕНИЕ:

Пусть $dp[i][mask]$ - минимальная стоимость распределить первых i работников, чтобы они выполнили множество заданий маски.

$$dp[i][mask] = \min_{b \in mask} (a_{ib} + dp[i-1][\underbrace{mask|b}_{mask+(1<b)}])$$

Асимптотика: $O(2^n n^2)$

1.1.2 Задача 2

Максимальная клика в графе за $O(2^{\frac{n}{2}})$

Определение 1.1. Клика $C \subset V$ такова, что для любых ее двух вершин есть ребро между ними.

Решим пока задачу за $O(2^k)$, где k - количество вершин

Обозначим за $neighbour(v)$ маску соседей v

Тогда $clique(mask) = true \leftrightarrow clique(mask|v) = true, mask|v \subset neighbour(v)$

Осталось только придумать, как из маски за $O(1)$ выкидывать ее вершину.

Сделаем это предсчетом для каждой маски за $O(2^k)$, записывая последовательно ее старший бит.

1.1.3 Задача 3

Найти максимальную клику в маске.

РЕШЕНИЕ:

1. Если $\text{clique}(\text{mask}) == \text{true}$, то $\text{subclique}(\text{mask}) = |\text{mask}|$
2. Возьмем максимальное значение из:
 - (a) $\text{subclique}(\text{mask} | v)$ - не берем v
 - (b) $1 + \text{subclique}(\text{mask} \& \text{neighbour}(v))$ - берем v

Такое тоже работает за $O(2^k)$

Теперь мы готовы решить основную задачу...

Шаг 1 Разобьем граф на 2 половинки, где будем искать клики. Пусть $\text{corr}[\text{mask}]$ - множество вершин правой доли, которые соединены со всеми вершинами mask .

Шаг 2 Хотим добавить их к mask , чтобы получилась клика. Единственное требование - все выбранные вершины $\text{corr}[\text{mask}]$ должны быть кликой \rightarrow А ЭТО ВЕДЬ ЗАДАЧА 3!!!
То есть ответ будет состоять из $\max(|\text{mask}| + \text{subclique}(\text{mask}))$, где mask - клика из левой части. Осталось понять, как считать $\text{corr}[\text{mask}]$

Шаг 0 $\text{corr}[\text{mask}] = \text{corr}[\text{mask}|v] \& \text{neighbour}(v)$

1.2 ДП по профилю

1.2.1 Задача 1

Пусть есть доска $n \times m$, сколько существует способов покрыть ее доминошками.

$dp[j][\text{mask}]$ - количество способов полностью покрыть j столбцов, т. ч. mask - множество строк, где лежат "торчащие" доминошки.

"Торчащие" доминошки - те, что расположены в j и $j + 1$ столбцах.

1. База: $dp[0][0] = 1, dp[0][\neq 0] = 0$
2. Переход: Обозначим за old_mask маску на $j - 1$ столбце. Переберем по всевозможным old_mask .
3. Заметим, что, зафиксировав mask и old_mask , картинка полностью заполняется. Добавляем $dp[i - 1][\text{old}]$ к $dp[j][\text{mask}]$, если
 - (a) $\text{old} \cap \text{mask} = 0$
 - (b) В $\text{old} \cup \text{mask}$ все блоки из нулей-четной длины

Получаем асимптотику $O(4^n m)$, но можно подправить на $O(3^n m)$, если не рассматривать случаи $\text{old} \cap \text{mask} = 1$ в каком-то бите.

1.3 Изломанный профиль

Считаем, что профиль - теперь часть доски, покрытая доминошками по предположению.

Комментарий: Раньше профиль получался из целых столбцов, а теперь нет