Лекция 3. Динамическое программирование

Екатерина Алексеева

Новосибирский Государственный Университет
Механико-математический факультет
http://math.nsc.ru/~alekseeva/

20.09.2012

Содержание лекции

Характеристики алгоритмов

Для оценки качества алгоритмов будем использовать два параметра:
T_A — трудоемкость (число элементарных операций алгоритма A);
Π_A — требуемый объем памяти.
Элементарная операция — одна из арифметических операций: сложение, вычитание, умножение, деление или логическая операция сравнение двух чисел.
Нас будет интересовать зависимость параметров алгоритма от длины записи исходных данных задачи с точностью до порядка величин.

Пример

При $= \frac{3}{2}n^2$, *будем писать* $T = O(n^2)$ *или* $T \approx n^2$.

Полиномиальные алгоритмы

Алгоритм A называют полиномиальным

Если его трудоемкость T_A ограничена полиномом от длины записи исходных данных, то есть существует константа $c > 0$ и натуральное число k такие, что $T_A \leq cL^k$, где L – длина записи исходных данных.

Алгоритм называют экспоненциальным

Если трудоемкость алгоритма нельзя ограничить полиномом от длины записи исходных данных.

Пример

Упорядочить n *чисел* $a_1, \ldots, a_n$ *в порядке неубывания.*
Алгоритм:
выбрать наименьший элемент, затем наименьший из оставшихся и т.д.
Длина записи исходных данных:

$$L = \sum_{i=1}^{n} log_2 a_i,$$

$$T = O(n^2) \leq O(L^2),$$

то есть алгоритм является полиномиальным.

Дано

n – число предприятий;
Y – количество единиц некоторого ресурса;
$f_k(x)$ – количество продукции, которое будет произведено на k-м предприятии, если в него будет вложено x единиц ресурса (монотонно неубывающая функция).

Требуется

максимизировать объем выпускаемой продукции

$$\max f_1(x_1) + \ldots + f_n(x_n) \tag{1}$$

при ограничениях:

$$x_1 + \ldots + x_n \leq Y, \tag{2}$$

$$x_i \geq 0 \text{ целые, } i = 1, \ldots, n. \tag{3}$$

Идея динамического программирования (ДП)

Разработчики: Р. Беллман, В.С. Михалевич, Н.З. Шор

Метод ДП можно трактовать как алгоритмическую версию рассуждений по индукции.

Пусть оптимальное значение целевой функции задачи (1)–(3)

$$S_k(y), 1 \leq k \leq n, 0 \leq y \leq Y,$$

где n заменено на k, Y заменено на y.

Требуется найти $S_n(Y)$ и набор переменных, на котором достигается это значение.

Теорема

Пусть $f_1(x_1), \ldots, f_n(x_n)$ – монотонно неубывающие функции. Тогда справедливы следующие рекуррентные соотношения:

$$S_1(y) = f_1(y), 0 \leq y \leq Y \tag{4}$$

$$S_k(y) = \max\{S_{k-1}(y-x) + f_k(x) | 0 \leq x \leq y\}, 2 \leq k \leq n, 0 \leq y \leq Y. \tag{5}$$

Доказательство

Соотношение (4) очевидно.
По определению $S_k(y) \geq \max\{S_{k-1}(y-x) + f_k(x) | 0 \leq x \leq y\}$. Пусть теперь $(x_1^\star, \ldots, x_k^\star)$ – такой вектор, что $x_1^\star + \ldots + x_k^\star \leq y$ и

$$S_k(y) = f_1(x_1^\star) + \ldots + f_k(x_k^\star).$$

Поскольку

$$S_{k-1}(y - x_k^\star) \geq f_1(x_1^\star) + \ldots + f_{k-1}(x_{k-1}^\star),$$

имеем $S_k(y) = f_1(x_1^\star) + \ldots + f_k(x_k^\star) \leq S_{k-1}(y - x_k^\star) + f_k(x_k^\star)$.

Алгоритм ДП вычисляет множество

$$S_k = \{S_k(y) | 0 \leq y \leq Y\}, k = 1, \dots, n$$

с помощью соотношений (4), (5), где на каждом шаге оптимизируется ровно одна переменная.

y	$S_1(y)$	$S_2(y)$	$\dots$	$S_n(y)$
0				
1				
2				
$\vdots$				
Y				$S_n(Y)$

Прямой ход алгоритма

Процесс вычисления $S_1, \dots, S_n$. *Трудоемкость:*

- *Число операций*

Алгоритм ДП вычисляет множество

$$S_k = \{S_k(y) | 0 \le y \le Y\}, k = 1, \dots, n$$

с помощью соотношений (4), (5), где на каждом шаге оптимизируется ровно одна переменная.

y	$S_1(y)$	$S_2(y)$	$\dots$	$S_n(y)$
0				
1				
2				
$\vdots$				
Y				$S_n(Y)$

Прямой ход алгоритма

Процесс вычисления $S_1, \dots, S_n$. *Трудоемкость:*

- *Число операций*
- $O(Y^2 n)$,
- *Память*

Алгоритм ДП вычисляет множество

$$S_k = \{S_k(y) | 0 \leq y \leq Y\}, k = 1, \ldots, n$$

с помощью соотношений (4), (5), где на каждом шаге оптимизируется ровно одна переменная.

y	$S_1(y)$	$S_2(y)$	$\ldots$	$S_n(y)$
0				
1				
2				
$\vdots$				
Y				$S_n(Y)$

Прямой ход алгоритма

Процесс вычисления $S_1, \ldots, S_n$. *Трудоемкость:*

- *Число операций*
- $O(Y^2 n)$,
- *Память*
- $O(Yn)$.

Обратный ход алгоритма

Вычисляются значения $(x_n^\star, \ldots, x_1^\star)$, *с учетом того, что уже известны* $S_k(y)$. *Например,* $x_n^\star$ *определяется из уравнения*

$$S_n(Y) = f_n(x_n^\star) + S_{n-1}(Y - x_n^\star)$$

и так далее. Трудоемкость:

- *Число операций*

Обратный ход алгоритма

Вычисляются значения $(x_n^\star, \ldots, x_1^\star)$*, с учетом того, что уже известны* $S_k(y)$*. Например,* $x_n^\star$ *определяется из уравнения*

$$S_n(Y) = f_n(x_n^\star) + S_{n-1}(Y - x_n^\star)$$

и так далее. Трудоемкость:

- *Число операций*
- $O(Yn)$,
- *Память*

Обратный ход алгоритма

Вычисляются значения $(x_n^\star, \ldots, x_1^\star)$*, с учетом того, что уже известны* $S_k(y)$*. Например,* $x_n^\star$ *определяется из уравнения*

$$S_n(Y) = f_n(x_n^\star) + S_{n-1}(Y - x_n^\star)$$

и так далее. Трудоемкость:

- *Число операций*
- $O(Yn)$*,*
- *Память*
- $O(Yn)$*.*

Характеристика алгоритма ДП для распределительной задачи

Пусть $f_i(x_i) = a_i x_i$, все коэффициенты a_i ограничены константой, $a_i \leq c$, но $Y = 2^n$. Тогда длина записи

$$L = \sum_{i=1}^{n} \log a_i + \log Y \leq c' n,$$

то есть $L = O(n)$, но

$$T = O(Y^2 n) = O(2^{2L} L),$$

и алгоритм ДП не является полиномиальным.

Обобщение распределительной задачи

Обобщим задачу (1)–(3):

$$\max f_1(x_1) + \ldots + f_n(x_n) \tag{6}$$

$$h_1(x_1) + \ldots + h_n(x_n) \leq Y \tag{7}$$

$$a_i \geq x_i \geq 0, \text{ целые, } i = 1, \ldots, n. \tag{8}$$

Если $h_i(x)$ целочисленные монотонно неубывающие функции, то вместо (4)– (5) можно использовать следующие рекуррентные соотношения:

$$S_1(y) = f_1(x^\star), \text{ где } x^\star = \max\{x \leq a_1 | h_1(x) \leq y\}, 0 \leq y \leq Y; \tag{9}$$

$$S_k(y) = \max_{x \leq a_k | h_k(x) \leq y} \{f_k(x) + S_{k-1}(y - h_k(x))\}, 2 \leq k \leq n, 0 \leq y \leq Y. \tag{10}$$

Упражнение. Доказать справедливость соотношений (9)–(10).

Обратная задача

Поиск наименьших затрат на получение заданного количества продукции:

$$\min h_1(x_1) + \ldots + h_n(x_n) \tag{11}$$

при ограничениях:

$$f_1(x_1) + \ldots + f_n(x_n) \geq D, \tag{12}$$

$$a_i \geq x_i \geq 0 \text{ целые, } i = 1, \ldots, n. \tag{13}$$

Если $f_k(x)$ – целочисленные монотонно неубывающие функции, то для решения задачи (11)–(13) можно использовать идеи динамического программирования.

ДП для обратной задачи

Пусть
$$f_i^{-1}(d) = \min\{0 \le x \le a_i | f_i(x) \ge d\}.$$
Для $1 \le k \le n$, $0 \le d \le D$ обозначим через $t_k(d)$ – оптимальное решение задачи (11)–(13), в которой n заменено на k, а D заменено на d.
Требуется найти $t_n(D)$.
Рекуррентные соотношения:

$$t_1(d) = \begin{cases} \infty, & \text{если } f_1(a_1) < d, \\ h_1(f_1^{-1}(d)), & \text{если } f_1(a_1) \ge d, \end{cases} \quad 0 \le d \le D. \qquad (14)$$

$$t_k(d) = \min\{t_{k-1}(d - f_k(x)) + h_k(x) | 0 \le x \le a_k, x \le f_k^{-1}(d)\}, k \ge 2, 0 \le d \le D. \qquad (15)$$

Упражнение. Доказать справедливость соотношений (14)–(15).

Теорема

Предположим, что D — наибольшее число, для которого оптимальное значение целевой функции задачи (11)–(13) не превосходит Y. Тогда оптимальное значение целевой функции задачи (6)–(8) равно D.

Доказательство

Пусть D удовлетворяет условию теоремы и $(x_1^\star, \ldots, x_n^\star)$ — соответствующее решение задачи (11)–(13). Значит $f_1(x_1^\star) + \ldots + f_n(x_n^\star) \geq D$ и $h_1(x_1^\star) + \ldots + h_n(x_n^\star) \leq Y$. Следовательно, D не превосходит оптимального решения D_1 задачи (6)–(8). Если бы $D1$ было больше D, то решение задачи (11)–(13), в которой D заменено на D_1, тоже не превышало бы Y, что противоречит максимальности D.

Рекомендуемая литература

- Т. Кормен, Ч. Лейзерсон, Р. Ривест Алгоритмы: построение и анализ. М.: МЦНМО, 1999
- М. Гэри, Д. Джонсон: Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982