Задача о рюкзаке

Динамическое программирование

Задача о ранце

- Общий вес ранца заранее ограничен. Какие предметы положить в ранец, чтобы общая полезность отобранных предметов была максимальна? Вес каждого предмета известен.
- Есть много эквивалентных формулировок. Например, можно вместо ранца рассматривать космический аппарат – спутник Земли, а в качестве предметов - научные приборы. Тогда задача интерпретируется как отбор приборов для запуска на орбиту. Правда, при этом предполагается решенной предварительная задача - оценка сравнительной ценности исследований, для которых нужны те или иные приборы.
- С точки зрения экономики предприятия и организации производства более актуальна другая интерпретация задачи о ранце, в которой в качестве «предметов» рассматриваются заказы (или варианты выпуска партий тех или иных товаров), в качестве полезности – прибыль от выполнения того или иного заказа, а в качестве веса – себестоимость заказа.

Математическая постановка

- Перейдем к математической постановке. Предполагается, что имеется n предметов, и для каждого из них необходимо решить, класть его в ранец или не класть. Для описания решения вводятся булевы переменные X_k, k = 1,2,…, n (т.е. переменные, принимающие два значения, а именно, 0 и 1). При этом $X_k = 1$, если предмет размещают в ранце, и $X_k = 0$, если нет, k = 1,2,…, n. Для каждого предмета известны две константы: A_k - вес k-го предмета, и C_k - полезность k-го предмета, k = 1,2,…, n . Максимально возможную вместимость ранца обозначим B. Оптимизационная задача имеет вид
- $C_1 X_1 + C_2 X_2 + C_3 X_3 + \ldots + C_n X_n \rightarrow \max$,
- $A_1 X_1 + A_2 X_2 + A_3 X_3 + \ldots + A_n X_n \leq B$.
- К целочисленному программированию относятся задачи размещения (производственных объектов), теории расписаний, календарного и оперативного планирования, назначения персонала и т.д.

Решить задачу о рюкзаке. Вместимость 9

i	1	2	3	4
$\mathbf{c_i}$	5	7	6	3
$\mathbf{q_i}$	2	3	5	7

Решить задачу о рюкзаке. Вместимость 7

i	1	2	3	4
$\mathbf{c_i}$	3	2	6	4
$\mathbf{q_i}$	5	3	5	3

Решить задачу о рюкзаке. Вместимость 7

i	1	2	3	4
$\mathbf{c_i}$	5	2	5	4
$\mathbf{q_i}$	6	3	5	3

Решить задачу о рюкзаке. Вместимость 8

i	1	2	3	4
$\mathbf{c_i}$	7	4	6	1
$\mathbf{q_i}$	5	1	3	5

Применение задачи о рюкзаке

- На основе задачи о рюкзаке в 1978 году Ральфом Мерклем и Мартином Хеллманом была разработана Ранцевая криптосистема Меркля-Хеллмана. Это была одна из первых криптосистем с открытым ключом, но, к сожалению, она оказалась криптографически нестойкой и, как следствие, не приобрела популярности.
- «Задача о рюкзаке» заключается в следующем: зная подмножество грузов, уложенных в ранец, легко подсчитать суммарный вес, но, зная вес, непросто определить подмножество грузов.
- В алгоритме шифрования не используются типы вещей, и потому результирующий вектор x содержит лишь 0 или 1.
- Р.Мерклю удалось получить обратную к числу s функцию, которая давала бы вектор x, зная только некий «секретный» ключ, и он предложил $100 тому, кто сможет раскрыть ранцевую систему Меркля-Хеллмана.
- Меркль использовал не произвольную последовательность w_i, а супервозрастающую, то есть такую, что $w_{k+1} > \Sigma w_i$, i=1,2,.., k.
- Шифрование
- – сообщение $x = (x_1, x_2, ..., x_n)$
- - вычисляем $y = b_1x_1 + b_2x_2 + \ldots + b_nx_n$

Пример шифрации

- w = {2, 7, 11, 21, 42, 89, 180, 354} - супервозрастающая последовательность.
- Она является основой для генерации закрытого ключа. Посчитаем сумму элементов последовательности. Она равна 706.
- Далее выберем простое число q, превосходящее полученное нами значение суммы. q = 881
- Выберем также число r из интервала [1,q) r = 588.
- Построим последовательность β, умножая каждый элемент из последовательности w на r по модулю q.
- 2 * 588 mod 881 = 235
- 7 * 588 mod 881 = 592
- 11 * 588 mod 881 = 301
- 21 * 588 mod 881 = 14
- 42 * 588 mod 881 = 28
- 89 * 588 mod 881 = 353
- 180 * 588 mod 881 = 120
- 354 * 588 mod 881 = 236
- Получим β = (295, 592, 301, 14, 28, 353, 120, 236).

Пример шифрования

- Пусть Алиса хочет зашифровать "a". Сначала она должна перевести "a" в двоичный код
- 01100001
- Далее она умножает каждый бит на соответствующее число из последовательности β, а сумму значений отправляет получателю.
- a = 01100001
- 0 * 235
- + 1 * 592
- + 1 * 301
- + 0 * 14
- + 0 * 28
- + 0 * 353
- + 0 * 120
- + 1 * 236
- = 1129

Расшифровка

- Чтобы расшифровать сообщение, Боб умножает полученное им значение на мультипликативное обратное r по модулю q.
- 1129 * 442 mod 881 = 372
- После этого Боб раскладывает 372 следующим образом. Сначала он выбирает наибольший элемент из w, который меньше, чем 372, и вычисляет их разность. Далее он выбирает следующий наибольший элемент, который меньше, чем полученная разность, и повторяет эти действия, пока разность не станет равной нулю.
- 372 - 354 = 18
- 18 - 11 = 7
- 7 - 7 = 0
- Элементы, которые были выбраны из w , будут соответствовать 1 в двоичной записи исходного текста.
- 01100001
- Переводя обратно из двоичной записи, Боб получает, наконец, искомое "a".