Лекция 5
Жадные алгоритмы

План лекции

1. Дискретная задача о рюкзаке.
2. Непрерывная задача о рюкзаке.
3. Задача о выборе заявок.
4. Коды Хаффмена.

Дискретная задача о рюкзаке
(пример решения частной задачи)

На складе имеется 5 неделимых предметов. Для каждого предмета известна его стоимость (в рублях) и масса (в кг). Величины стоимости и массы являются натуральными числами. Ваша цель состоит в том, чтобы определить максимальную суммарную стоимость предметов, которые можно унести со склада при условии, что суммарная масса предметов не должна превышать 16 кг.

Пусть элемент C_i таблицы C соответствует стоимости i-го предмета, а элемент M_i таблицы M – массе i-го предмета. Будем считать, что предметы пронумерованы в порядке их следования в таблицах.

Пусть T обозначает функцию, значение которой соответствует решению нашей задачи. Аргументами функции является количество предметов (по этому аргументу можно определить их стоимости и массы соответствующих предметов), а также максимальная суммарная масса, которую можно унести.

Определим подзадачи $T(i, j)$, где i обозначает количество начальных предметов, из которых можно осуществлять выбор, а j определяет максимально возможную суммарную массу уносимых предметов. Отметим, что определенный таким образом первый параметр i определяет как количество предметов для подзадачи, так и значения их стоимостей и масс, определяемых из таблиц C и M.

Определим сначала начальные значения функции T. При нулевых значениях одного из аргументов значение функции равно нулю:

$T(0, 0) = 0$

$T(0, j) = 0$ при $j \geq 1$,

$T(i, 0) = 0$ при $i \geq 1$.

Определим возможные значения функции $T(i, j)$ при ненулевых значениях аргументов.

Решение подзадачи, соответствующей функции $T(i, j)$ может быть сведено к двум возможностям: уносится ли при наилучшем решении предмет с номером i или нет.

Если предмет не уносится, то решение задачи с i предметами сводится к решению подзадачи с $i - 1$ предметами, т. е.

$$T(i, j) = T(i - 1, j).$$

Если предмет с номером i уносится, то это уменьшает максимально возможную суммарную массу для $i - 1$ первых предметов на величину $M[i]$, одновременно при этом увеличивая значение решения для оставшихся предметов $T(i - 1, j - M[i])$ на величину $C[i]$, т. е.

$$T(i, j) = T(i - 1, j - M[i]) + C[i].$$

При этом необходимо учитывать, что вторая ситуация возможна только тогда, когда масса i-го предмета не больше значения j.

Для получения наилучшего решения необходимо выбрать лучшую из этих двух возможностей. Поэтому рекуррентное соотношение при $i \geq 1$ и $j \geq 1$ имеет вид

$T(i, j) = T(i - 1, j)$ при $j < M[i]$ и

$T(i, j) = max(T(i - 1, j), T(i - 1, j - M[i]) + C[i])$ при $j \geq M[i]$.

Пусть заданы следующие значения стоимости и массы для 5 предметов:

$C[1] = 5, M[1] = 4$; $C[2] = 7, M[2] = 5$;

$C[3] = 4, M[3] = 3$; $C[4] = 9, M[4] = 7$;

$C[5] = 8, M[5] = 6$.

Таблица значений функции T, которую мы также назовем T, выглядит следующим образом:

$i \backslash j$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	5	5	5	5	5	5	5	5	5	5	5	5	5

2	0	0	0	0	5	7	7	7	7	12	12	12	12	12	12	12	12
3	0	0	0	4	5	7	7	9	11	12	12	12	16	16	16	16	16
4	0	0	0	4	5	7	7	9	11	12	13	14	16	16	18	20	21
5	0	0	0	4	5	7	8	9	11	12	13	15	16	17	19	20	21

Следовательно, решение задачи $T(5, 16) = 21$. Однако это не значит автоматически, что суммарная масса уносимых предметов равна 16. Действительно, $T(1, 16) = 5$, а соответствующая этой подзадаче суммарная масса предметов равна 4. Рассмотренная задача является широко известной задачей о рюкзаке.

```
T[0, 0]: = 0
 for j:= 1 to 16 do
 T[0, j] = 0;
 for i:=1 to 5 do
 T[i, 0]: = 0
 For  i:=1 to 5 do
    for j:=1 to 16 do
    if  j >= M[i] then
         T[i, j]: = max(T[i – 1, j], T[i – 1, j – M[i]] + C[i])
        else
         T[i, j] = T[i – 1, j]
```

<u>Общая формулировка дискретной задачи о рюкзаке</u>

Написать программу для задачи о рюкзаке, позволяющую определять набор уносимых предметов, когда заданы следующие параметры:

N – количество имеющихся предметов;

V - максимально возможная суммарная масса уносимых предметов;

C[1..N] - матрица стоимостей предметов;

M[1..N] – матрица масс предметов.

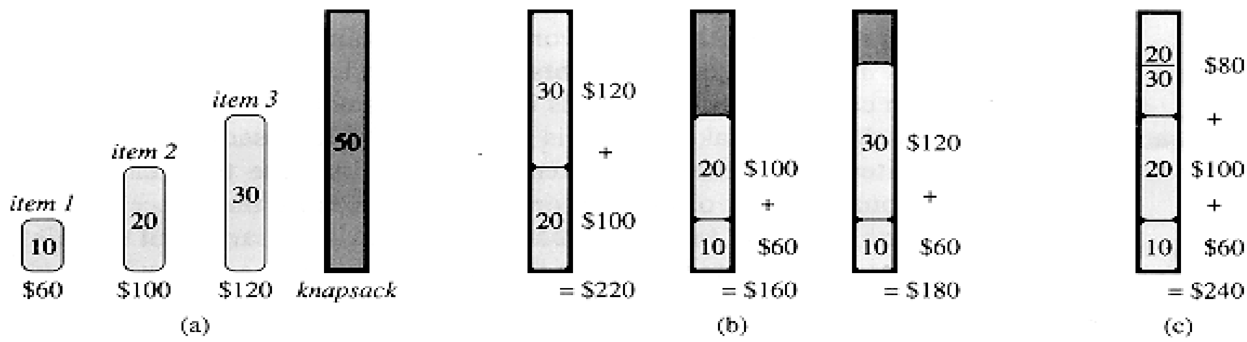
Непрерывная задача о рюкзаке

Задача отличается от дискретной тем, что можно дробить предметы на части и укладывать эти части в рюкзак (так, если в дискретной задаче речь идет о золотых слитках, то в непрерывной – о золотом песке). Обе задачи о рюкзаке обладают свойством оптимальности для подзадач. Для непрерывной задачи о рюкзаке существует более простой и быстрый алгоритм, называемый **жадным.** Этот алгоритм делает на каждом шаге локально оптимальный выбор, Это не всегда так – но для многих задач такой алгоритм дает оптимум.

Решим непрерывную задачу о рюкзаке с помощью жадного алгоритма. Сначала вычислим цены всех предметов - стоимость предмета разделим на массу. Сначала берем самый дорогой предмет и наполняем им рюкзак, если предмет закончился, а рюкзак не заполнен, берем следующий по цене, и так далее, пока не наберем максимально возможную суммарную массу предметов.

При решении этой задачи реализован принцип **жадного выбора** - когда последовательность локально оптимальных выборов дает глобально оптимальное решение

Для дискретной задачи о рюкзаке жадный алгоритм может оказаться неправильным



На рисунке показана задача выбора двух предметов из трех так, чтобы их суммарный вес не превысил 50 кг, жадный алгоритм дает неоптимальное решение. На последнем рисунке – решение аналогичной непрерывной задачи.