

## Динамическое программирование.

- Поиск экстремума
- Бит. поиск.

## Нарядное программирование

Принцип:

- 1) Оптимизация решения.
- 2) Оптимальное решение подзадачи

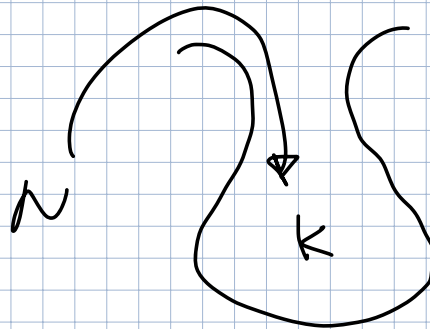
В подзадаче

Оптимальное решение задачи.

## Задача 1 (Задача о рюкзаке)

Тривиально:

$$T(N) = O(C_N^k)$$



## Улучшаем

Вместимость:  $W$

Вещи:

$w_1; w_2; w_3; \dots w_n$

Берем  $\min \rightarrow \min \text{ оставш } \rightarrow \dots$

Алгоритм:

→ отсортировать по возрастанию

→ Берем минимальный.

пока  
элемент не закончился.

Несложно:

```
arr, W  
count = 0  
summ = 0  
  
 $O(N \log N)$  → arr = sort(arr)  
for i = 0... N:  
    если arr[i] + summ ≤ W:  
        summ += arr[i]  
        count ++  
    иначе break
```

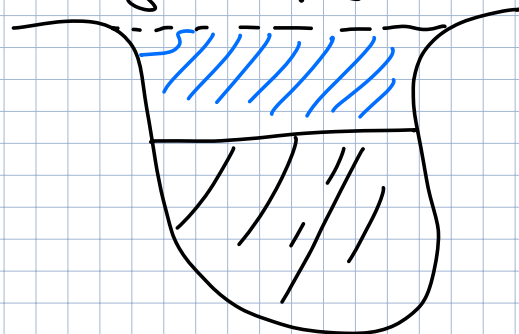
$N \cdot O(1) = O(N)$        $O(1)$



Асимптотика:  $T(N) = O(N \log N)$

1.2

Максимальный вес предмета  
при условии веса рюкзака.



```
arr, W  
count = 0  
summ = 0  
  
arr = sort(arr)  
biggest_elem = -1  
for i = 0... N:  
    если arr[i] + summ ≤ W:  
        summ += arr[i]  
        count ++  
        biggest_elem = arr[i]  
    иначе break
```

summ = biggest\_elem

```

for i = id(biggest_elem) ... N:
    если arr[i] + sum <= W:
        biggest_elem = arr[i]
    иначе break
print (sum + biggest_elem)

```

написать  
последний  
элемент.

Задача 2 (Задача о Рюкзаке  
Задача о Весе)

кон-во предметов  $N$

различные  $\checkmark$  веса предметов  $W$

	0	1	2	3	4	5	6	7	...
0	0	0	0	0	0	0	0	0	...
1	0	0	0	0	7				
2	0	0	3	3	7				
3	0	0	3	6	10				
4	0	1	3	6	13				
5	0	1	3	9	13				
6	0	1	4	9	16				
W	7	0	1	4	9	16	7+		

нельзя  
поменять  
положить

текущий  
элемент

столбец: 16  
вес: 6

ни о чем  
веса не  
попадают

КВАРТИРА	
вес ( $w_i$ )	стоимость ( $c_i$ )
4	1
2	3
3	6
1	7

ПСЕВДОКОД

table = [N][W]

- массив размера  $N \times W$

// i - столбец  
// j - строка

flat[0]  $\rightarrow$  вес  
flat[1]  $\rightarrow$  стоимость

for  $i, j = 1 \dots N \times W$   
 $\hookrightarrow \text{table}[i][j] = 0$

for  $i = 1 \dots N$ : ← порядок кон-ва вершин

for  $j = 1 \dots W$ :  
 $\text{table}[i][j] = \text{table}[i-1][j]$

вес  $i \rightarrow j$   
 весу

если  $\text{flat}[i][0] \leq j$   
 $t = \text{table}[i-1][j - \text{flat}[i][0]] + \text{flat}[i][1]$

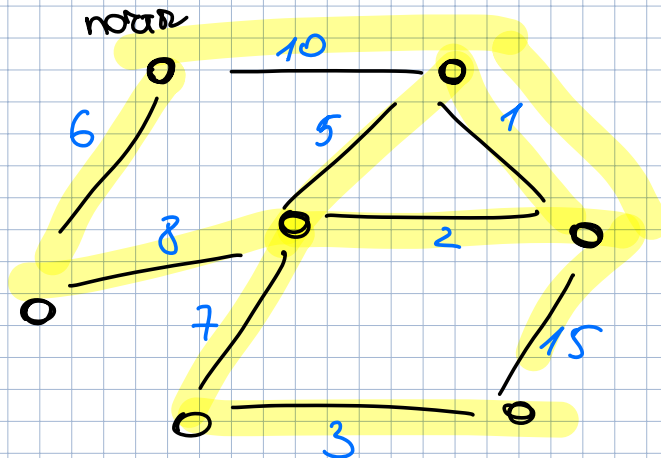
если  $t > \text{table}[i][j]$   
 $\hookrightarrow \text{table}[i][j] = t$

} }

Осталось пройти по построенной таблице!  
 и найти оптимальный порядок.

Задача 3

Задача Конн-Вольфа



Оптимальный маршрут

$O(E+V)$   
 $O(E+V)!$