

# Динамическое программирование

D&C

Оп.  $G$  граф:

$V$  - множество

$E$  - множество ребер

1) неор. граф  $E = \{ [i, j] \mid i, j \in V \}$

2) ор. граф  $(i, j)$



DAAG - Directed acyclic graph



Оп. топ. сорт.

или "линейная запись"

DAAG не может содержать  $v_1, v_2, \dots, v_n$  что  $(v_i, v_j) \in E$



УТБ. 1) граф  $G$  - DAG

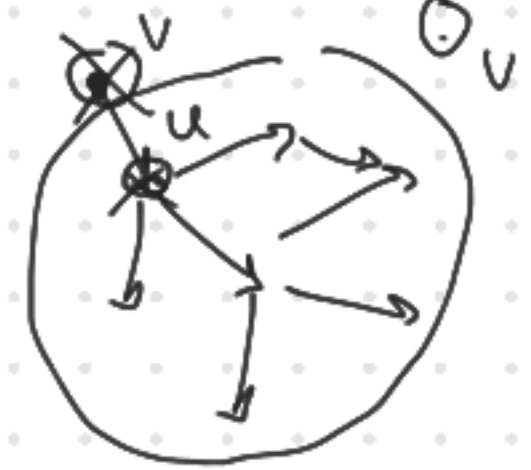
2)  $G$  имеет топ. сорт.

Доказ.

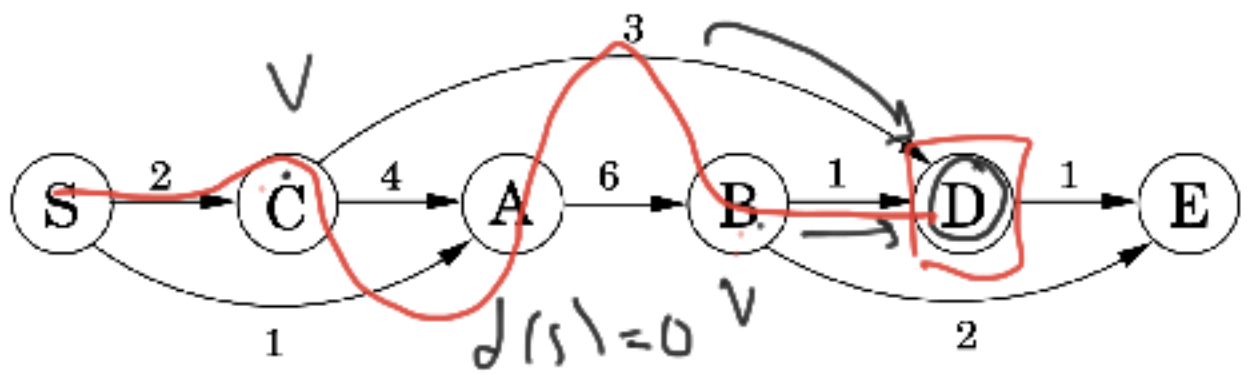
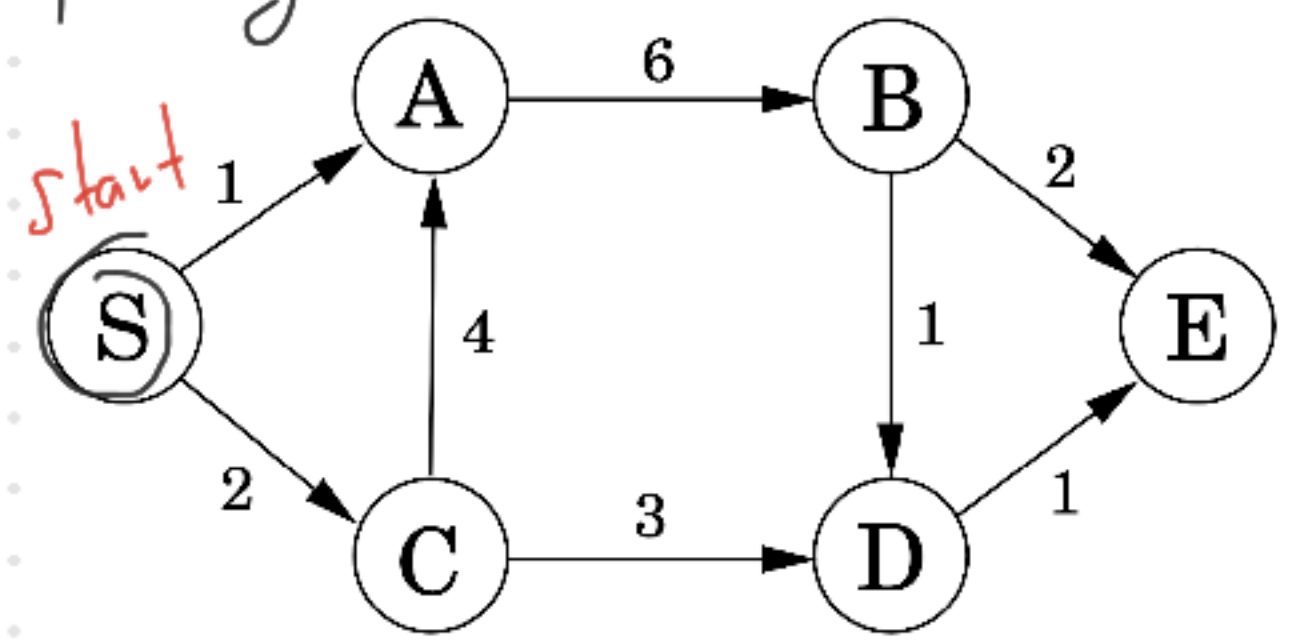
$2 \Rightarrow 1$  Пусть  $2$  и нег.



$1 \Rightarrow 2$



Зр. путь в DAG



$d(v) = \text{dist}(S, v)$  — кратчайший путь

$d(v) = \min_{u: (u,v) \in E} \{d(u) + l(u,v)\}$

$O(|V| + |E|)$

$O(h+m)$

max

min

Задача на G - DAG с given нач. фидей  
найти кратчайший путь  $S \rightarrow v_i$

- D.P. "dynamic programming"
- 1) initialize  $d(v)$
  - 2) работай на воз.  $d(v) = \dots$
  - 3) DAG работай
  - 4) работай отсюда
  - 5) работай отсюда  $d(S) = 0$

graph. непереход  $d(v)$

programming linear

initialize all  $\text{dist}(\cdot)$  values to  $\infty$   
 $\text{dist}(s) = 0$   
 for each  $v \in V \setminus \{s\}$ , in linearized order:  
 $\text{dist}(v) = \min_{(u,v) \in E} \{\text{dist}(u) + l(u,v)\}$

Науд. log setaony. moy wold

l.i.s.

$a_1, a_2, \dots, a_n$

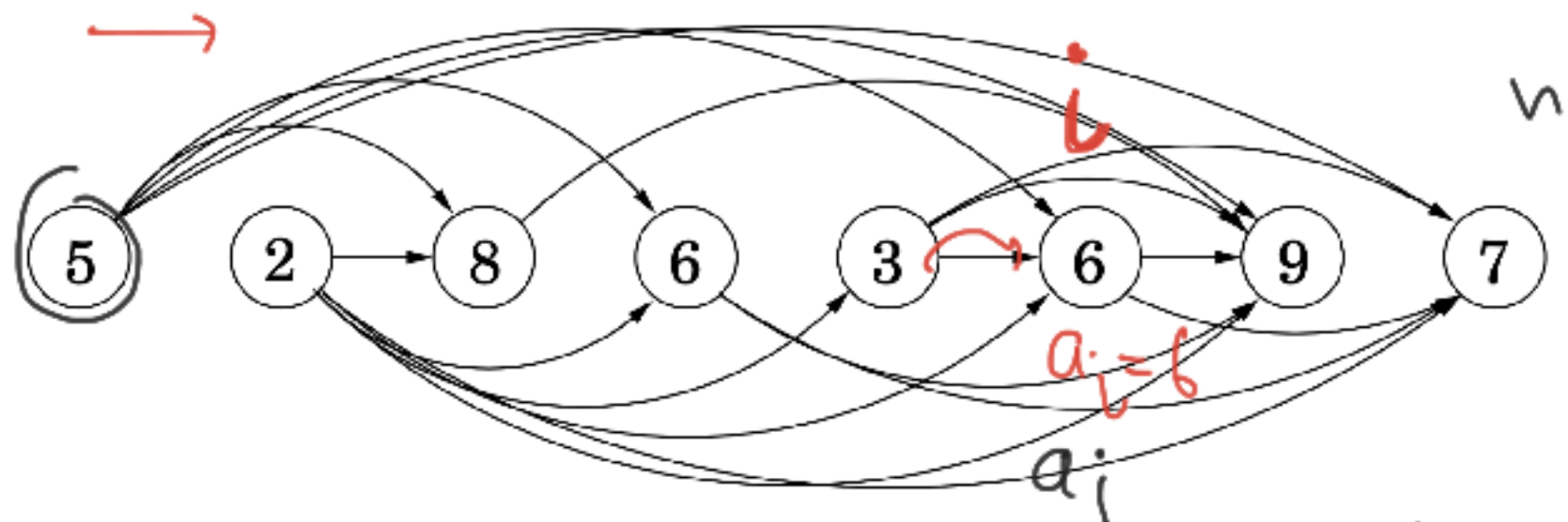
$a_{i_1} < a_{i_2} < a_{i_3} \dots a_{i_k}$   $k = \max$

1) найти  $k$

2) найти  $\{a_{ij}\}$

$b < c$   
 $i_b < i_c$

$\boxed{5} \boxed{2} 8 6 \boxed{3} \boxed{6} \boxed{9} 7$   $k=4$



$L(v) = \max_{u \rightarrow v} L(u) + 1$   
BT let  $\max_v L(v)$

Бажа  $h=0$

Примы  $h \rightarrow h+1$

$2^k \rightarrow 2^{k+1}$

$$L(i) = \max_{\substack{j < i \\ a_j < a_i}} (L(j) + 1)$$

$O(n^2)$

можно  $O(n \log n)$

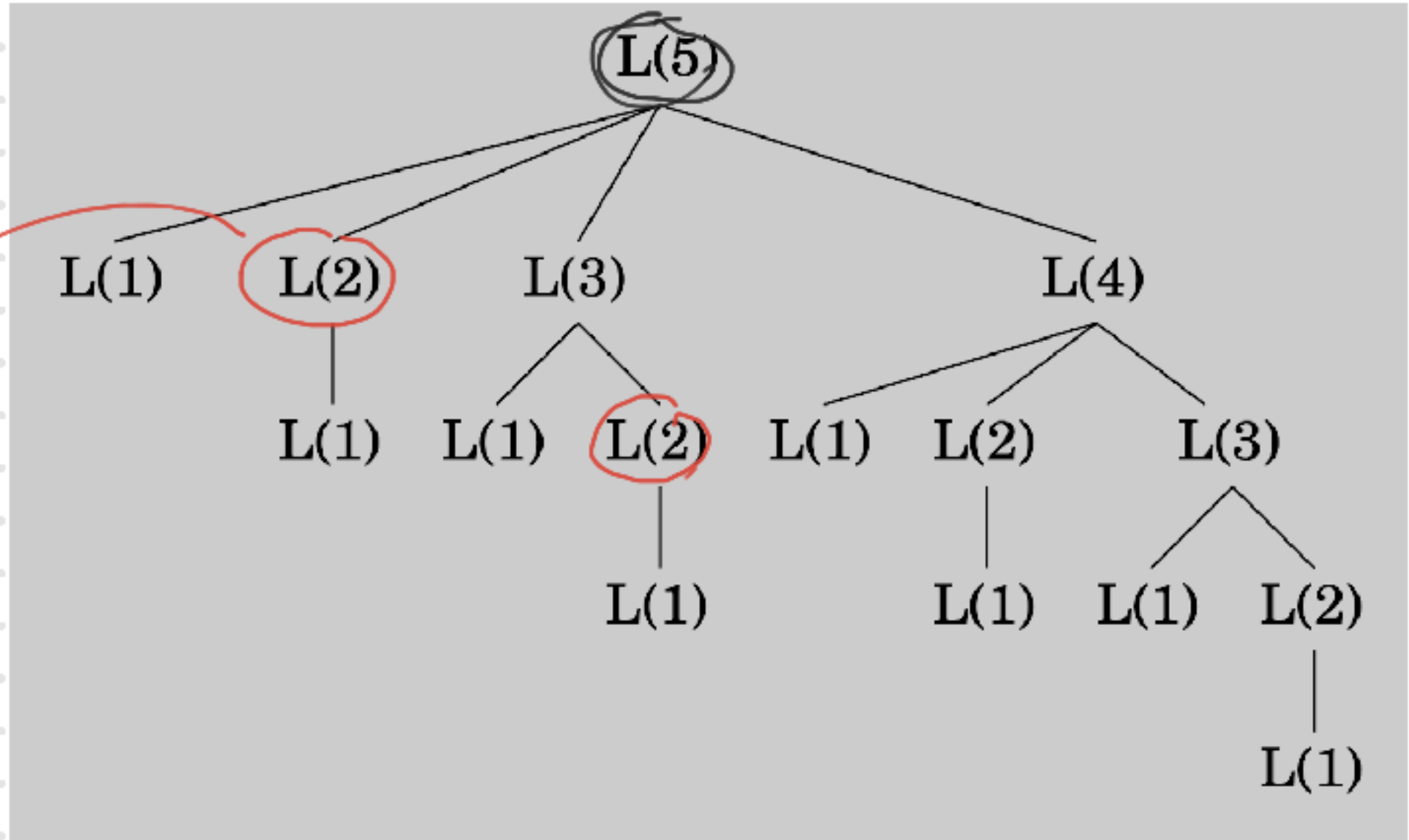
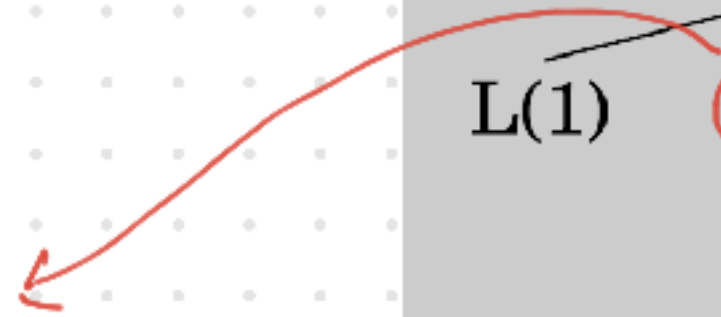
for  $j = 1, 2, \dots, n$ :  
 $L(j) = 1 + \max\{L(i) : (i, j) \in E\}$   
return  $\max_j L(j)$

$a_b \sim a_j \sim a_k$

DP vs Recursion

Мемоизация

w: L(w)



Fib: S



Показатель	1	2	...	n
веса	$w_1$	$w_2$	...	$w_n$
значения	$v_1$	$v_2$	...	$v_n$



найти  $I \subseteq [n]$

$$\sum_{i \in I} w_i \leq W$$

$$\sum_{i \in I} v_i \rightarrow \max$$

Динамическое программирование -  
 - работающая за  
 $\text{poly}(n \cdot W)$

0) первый шаг



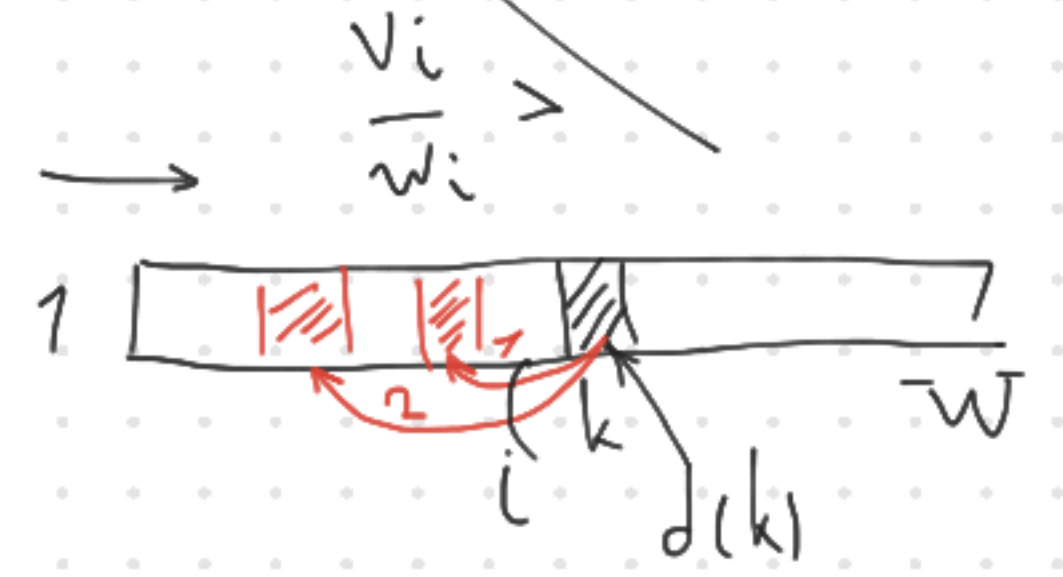
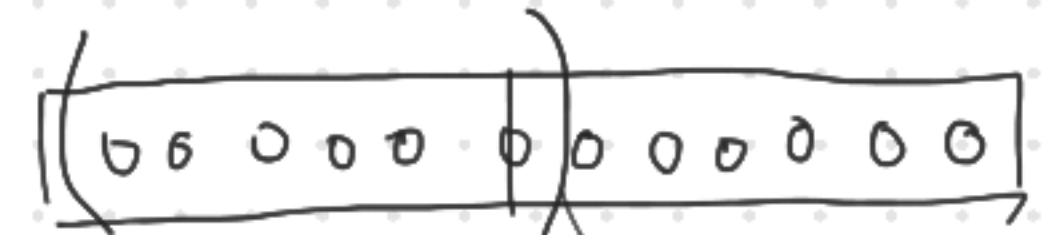
1) гурки: объект или сумма  
 с повторами

$d(k)$  - max value for  
 рюкзака вместимости k

$$d(k) = \max_{1 \leq i \leq n} (d(k - w_i) + v_i)$$

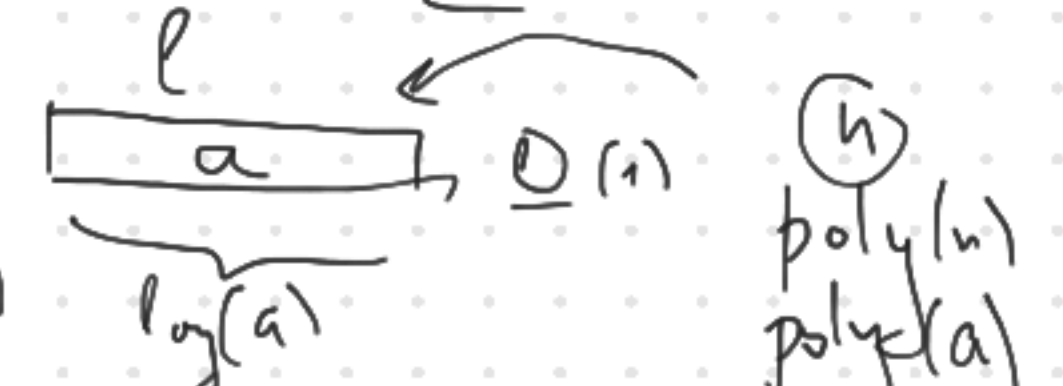
W малое, n очень много  
 размер массива  $W - \log W$

$$\sum a_i x^i = f(x)$$



$$O(n \cdot W) - \text{форма}$$

$$O(W) - \text{на месте}$$



$a \in \mathbb{N}$

$\text{poly}(n)$   
 $\text{poly}(a)$

2) гурк. Дег холторов

$w_1, w_2, \dots, w_n \in \mathbb{N}$   
 $v_1, v_2, \dots, v_n \in \mathbb{R}_{\geq 0}$

$d(k, i)$  - max value функце  
 мест. k  
 функциево д'фарт танко  
 1 - i н'ф'метри

$$d(k, i) = \max \{ d(k, i-1), d(k - w_i, i-1) + v_i \}$$

$d(w, w)$  - answer

Time:  $O(kW)$   
 space:  $O(kW) \rightarrow O(w)$   
 Дег боост. отбета

NP-hard proof complexity  
 fine-grained

SAT satisfiability  
 $\varphi = (x_1 \vee x_2) \wedge x_3 \vee x_4$   
 $x \in \{T, F\}$   
 $\exists$  м'ф'гт  $\{x_i\} : \varphi = T$   $2^{n(1-\epsilon)}$   
 ETH:  
 SAT + f'f'g'g'g'  $2^{n(1-\epsilon)}$

