

# Динамическое программирование

①

Задача  
└─┬─  
  └─  
  подзадачи

Merge Sort

1. Решить подзадачу  
внезапно решить задачу.

- возрастающая дитаминка
- нисходящая
- левый рекурсивный.

②

Числа Фибоначчи

$$f_0 = 0$$

$$f_1 = 1$$

$$\forall n > 1$$

$$f_n = f_{n-1} + f_{n-2}$$

$$f_2 = f_1 + f_0 = 1 + 0 = 1$$

$$f_3 = f_2 + f_1 = 1 + 1 = 2$$

$$f_4 = f_3 + f_2 = 2 + 1 = 3$$

$$f_5 = f_4 + f_3 = 3 + 2 = 5$$

.....

Функция  
}

Fib ( n )

если  $f[n] \neq -1$   
↳ return  $f[n]$

если  $n = 0$   
↳ return 0

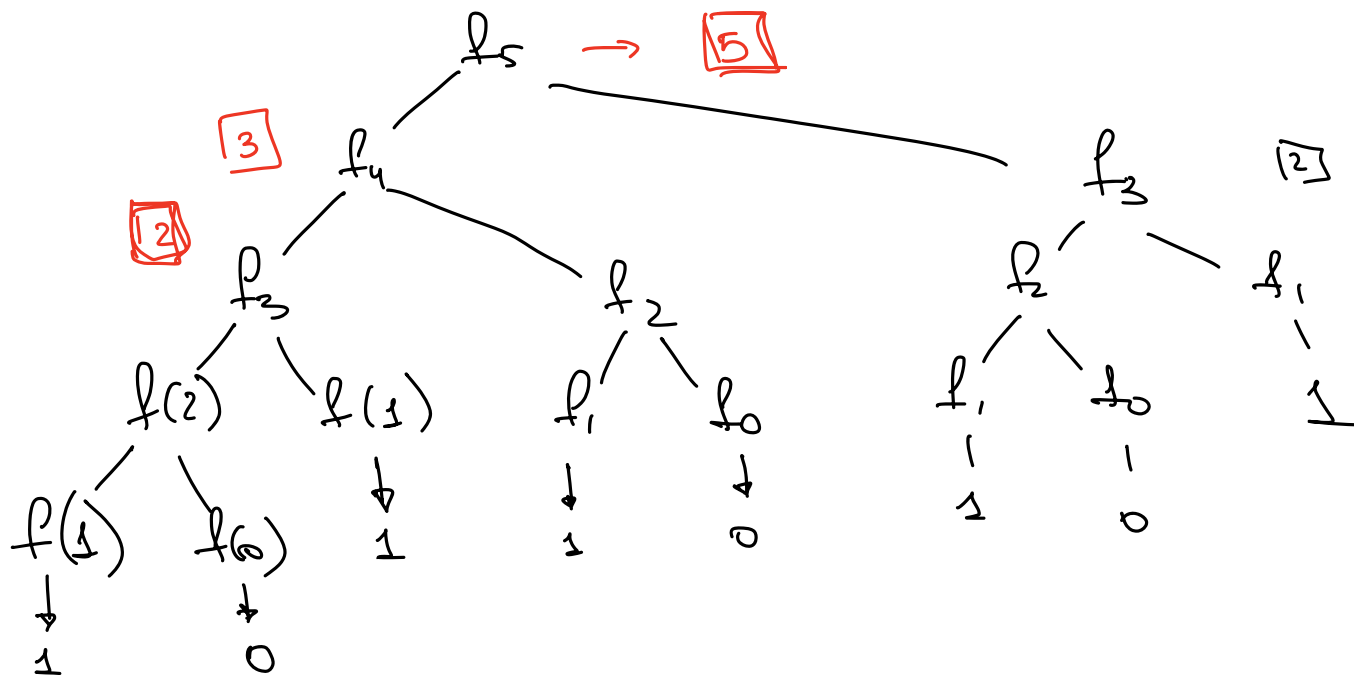
если  $n = 1$   
↳ return 1

return  $f(n-1) + f(n-2)$

}

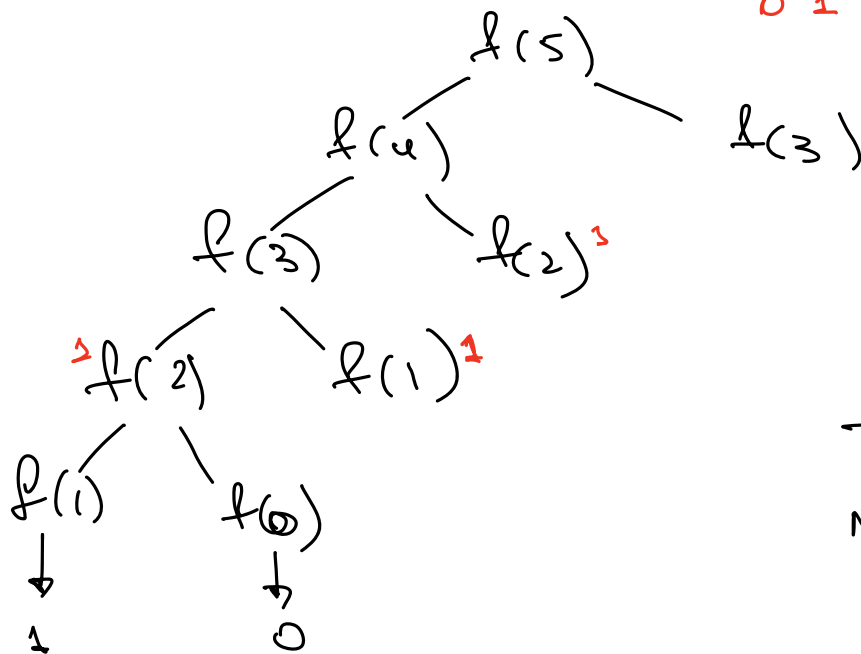
f10  
print(fib(10))

↓ нисходящая  
рекурсия.



Число вызовов  $f(5)$

0	1	2	3	4	5
<del>0</del>	<del>1</del>	<del>2</del>	<del>3</del>	<del>4</del>	<del>5</del>
0	1	1	2	3	5



$$T(N) = O(N)$$

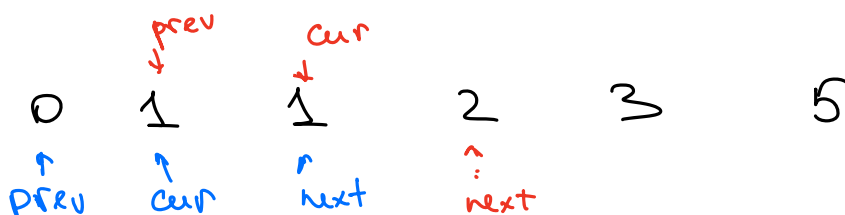
$$M(N) = O(1)$$

Углубление:

$$T(N) = O(N)$$

$$M(N) = O(1)$$

$$f_n = f_{n-1} + f_{n-2}$$



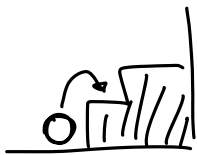
```

prev = 0
cur = 1
for i = 1 ... N:
    next = prev + cur
    print(next)
    prev = cur
    cur = next

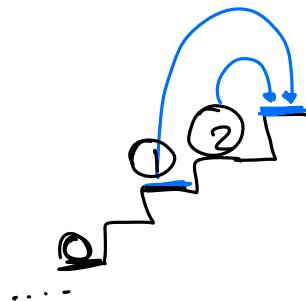
```

III Задача о кызгаче

Вариант 1



+1  
+2



3

Steps[0] = 1

Steps[1] = 1

Steps[n] = Steps[n-1] + Steps[n-2]

IV Бинарно пошук

а) Линеарно пошук

arr = [ ... ]

key?

Ему

ну

решо

key

&

array?

flag = false

for i = 0 ... arr.size

```

2
    if arr[i] == key
        flag = True
        break
}

```

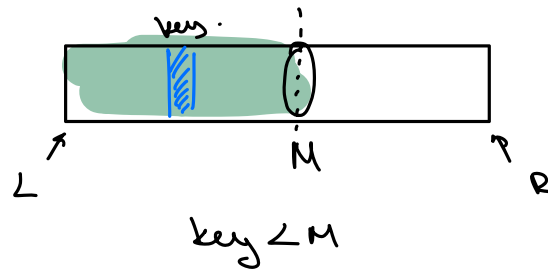
Время?  
 $O(N)$

```

if flag == True:
    print(name)
else:
    print("no name")

```

⑧ Бинарный поиск  
 - метод поиска в массиве



оп-гкыс

binSearch (arr, key)

```

L = 0
R = arr.size()
while L + 1 < R
{

```

$$M = \frac{L + R}{2}$$

если  $arr[M] > key$ :

$R = M$

если  $arr[M] < key$

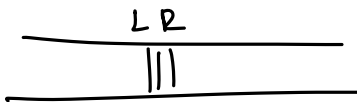
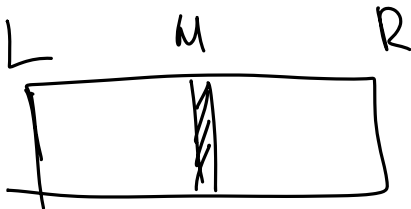
$L = M$

иначе

return True

}

return  $arr[L] == key$  or  
 $arr[R] == key$



$$\frac{2}{2} \frac{2}{2}$$

$$\frac{N}{2^k} = 1$$

$$N = 2^k$$

$$k = \log_2 N$$

Kyoto.

O(N)

$$N = 10^9$$

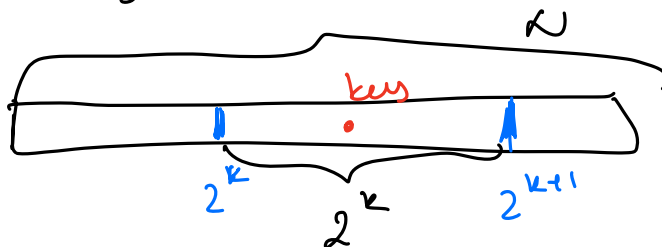
$$\Rightarrow \sim 10 \text{ cm}$$

$$\underline{O}(\log N) \Rightarrow$$

$$\frac{27}{10^8} \text{ cm}$$

②

Экспотенуальное поск



$$2^{k+1} - 2^k = 2 \cdot 2^k - 2^k = 2^k$$

for  $i = 0 \dots \log N - 1$

ecm  $arr[2^i] \leq key \leq arr[2^{i+1}]$

↳  $L = 2^i$   
 $R = 2^{i+1}$   
 break

$$\underline{O}(\log i)$$

bin search (arr, key, l, r)

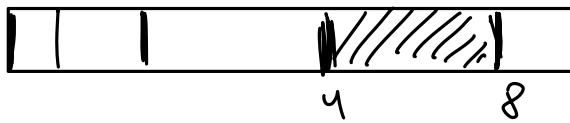
Бит. нонсе

$$O(\log N)$$

$\Delta K_{cn}$  nonch

$$\underline{O}(\log i)$$

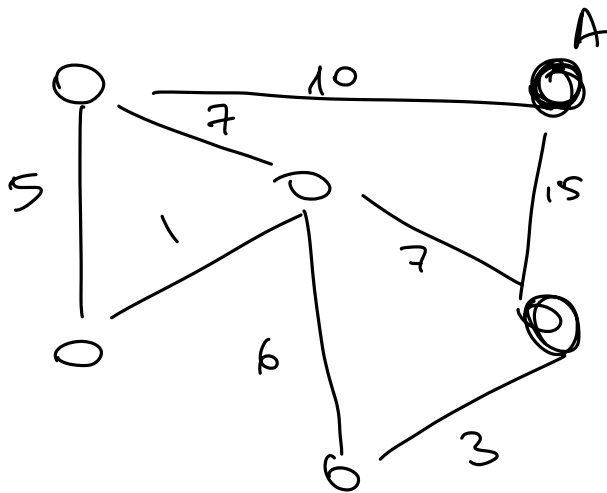
↑  
urgenc  
key.



$$2^k \geq i < 2^{k+1}$$

$$i = \log_2 k$$

⑤ Ζαγάρα ο Κονιβολιάρης



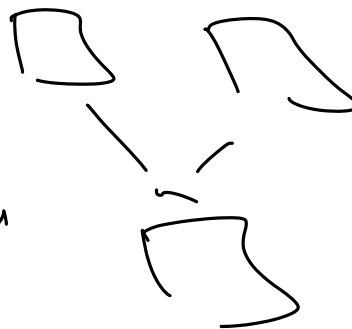
NP-πληρες  
πληρες  
πληρες

100000

⑥ Κλασική αντοχή.

η τωρινη

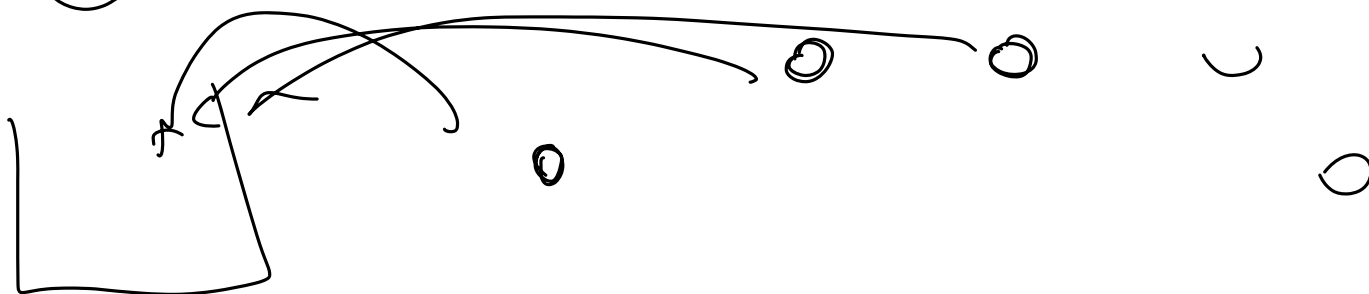
πικραει κατ' εξακριβω  
6 παρ' εκδρα, πικρο  
ικ κομμοζαει  
ζαει ανημαλοτοι  
ανημα.



VII

Задача

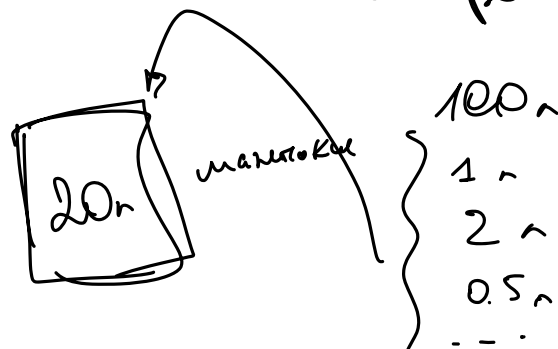
Купюромате



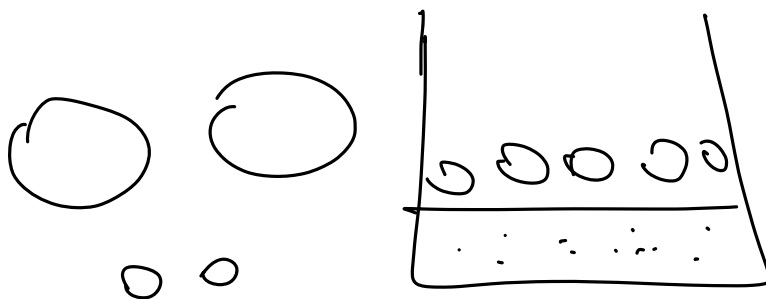
макс

кон-во  
банеж.

Как решить?



1. Сортируем по возрастанию
2. Берем банкноты из набора, пока не заполним.



```
arr = [ ... ] // все купюры.  
W - вместимость рюкзака  
arr.sort()  
count = 0 // кон-во купюр  
for i = 0..arr.size  
    if W >= arr[i]:  
        W -= arr[i]  
        count += 1
```

Print (count)

VIII

Упражнение.



Плато



Угол

В

$dp[1] = "w"$

$dp[2] = "w"$

} тем же образом

for  $i = 3 \dots N$ :

if  $dp[i-1] = "L"$  or  $dp[i-2] = "L"$

$dp[i] = "w"$

else:

$dp[i] = "L"$

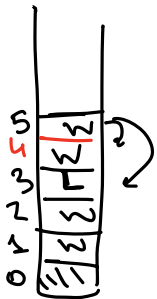
если  $dp[N] = "L"$

↳ В

иначе

↳ Плато.

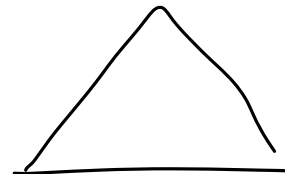
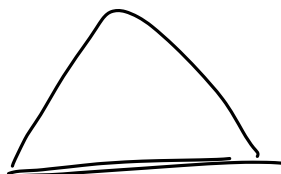
отсюда  
суждение.



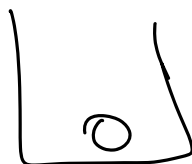
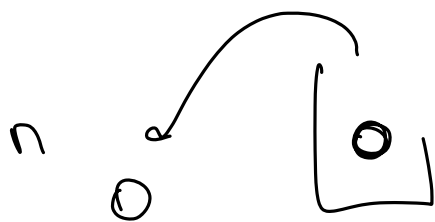
Модификация упражнения.

Угол

1 блок







B

$$dp[0][1] = w$$

$$dp[1][0] = w$$

$m \leftarrow \text{size\_array}$

	0	1	2	3	4	5
0		w	w	L		
1	w	L	w	w		
2	w	w	L	w		
3	L	w	w	L		
4						
5						
6						
⋮						

$N \uparrow \text{size\_array}$

for  $i = 0 \dots N$ :

for  $j = 0 \dots m$ :

flag = "L"

ecm

flag = "w"

$i - 1 \geq 0$

flag = "w"

// - gas j

$dp[i][j] = \text{flag}$ .

← глобальный массив.

(1x)

Задача

o

прокзак

Значение.

N - кол-во снушков

ber  
PtoK 3gK2

$$\begin{aligned} 1 &\rightarrow 2 \text{ kr} \\ 2 &\rightarrow 1 \text{ kr} \end{aligned}$$
$$dp[i][j] = dp[i][j-1]$$

Век прожиза  
↓  
дез нощу рену  
Ж-та

$$\rightarrow dp[i][j] = dp[i - w[j]][j-1] \text{ or } dp[i][j-1]$$

for  $i = 0, \dots, n$

for  $j = 0 \dots N$ :

ecnu  $i > w[i]$

$$\hookrightarrow dp[i][j] = dp[i - w[j]][j - 1] \text{ or } dp[i][j - 1]$$

уфа

$$dp[i][j] = dp[i][j-1]$$