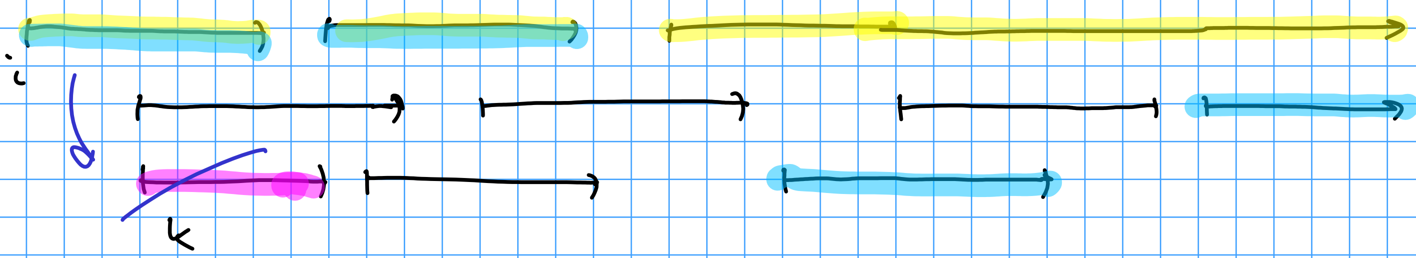
Жадные алгоритмы

Задача о выборе заявок

$[12{:}00,\ 13{:}40)$ лекция по алгоритмам
$[10{:}00,\ 18{:}00)$ конф. по нанотехнологиям
$[14{:}00,\ 15{:}40)$ практика по алгоритмам
$\vdots$
$[s_i,\ f_i)$

<u>Цель</u>: выполнить наибольшее кол-во заявок



Утв: Заявка с $\min f_i$ будет входить в одно из оптимальных решений.

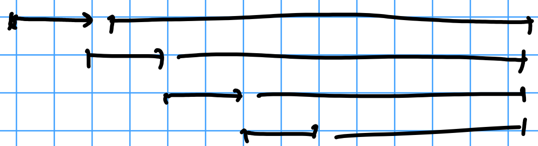
$\triangleright$ $\angle$ некоторый оптимальный набор заявок S.
$\exists\ f_i - \min$.
$i \notin S$

Возьмём первую заявку из S (с $\min f_k$)
$(S \setminus \{k\}) \cup \{i\}$ — это будет решение и причём оптимальное.
$f_k \geq f_i \Rightarrow$ это решение $\triangleleft$

Следствие: жадный алгоритм даёт опт. решение.

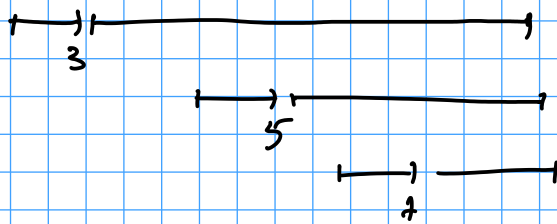
```
Tickets({s_i, f_i})
    Sort({s_i, f_i} по f_i)        O(n log n)
    F = 0
```

```
for (s, f) по возр. f:
    if s ≥ F:
        AddToSolution(s, f)
        F = f
```



(3 5 7 10)

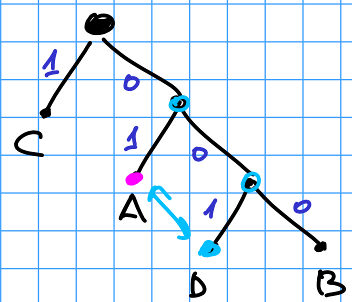
↓



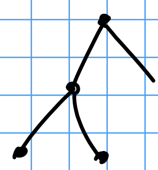
Код Хаффмана

A	30	00	01
B	10	01	000
C	70	10	1
D	5	11	001
	$\Sigma = 230$		175

prefix-free



Утв: На нижнем уровне в оптимальном дереве ≥ две вершины.



$2 \cdot f_i + 3 \cdot f_j \geq$

$f_i \leq f_j$

$\geq 3 f_i + 2 f_j$

c_i – символ i

f_i – частота c_i в тексте

$$C(T) = \sum f_i \cdot d(c_i) \to \min$$

Идея

$(a_1 \leq a_2 \leq a_3, \ldots, a_n)$ (3 5 7)

$(b_1 \leq b_2 \leq \ldots \leq b_n)$ (2 4 8)

$\sum a_i b_k$ min $\sum a_i b_{n-i+1}$

$$C(T) = \sum f_i \cdot d(c_i) = \sum f(n)$$

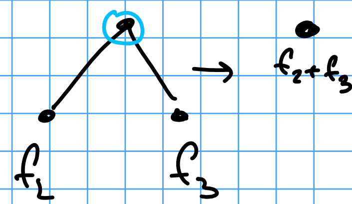
n - узел T кроме корня.

Утв: Символ с min f_i - на нижнем уровне.

Утв: Символ со вторым по значению f_i - на нижнем уровне.

1. f_1 $\quad f_2 \quad f_3$ ..... f_n
2. f_1 $\quad (f_2+f_3)$ ..... f_n

Утв. Оптимальное дерево для ① - это оптимальное для ②



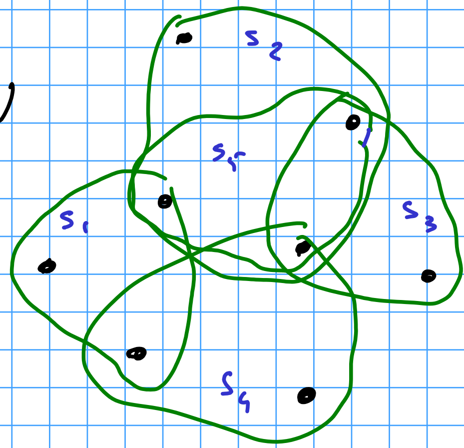
$$C(T_1) = C(T_2) + (f_2+f_3)$$

Huffman($\{c_i, f_i\}$):

```
PQ ← Make_Priority_Queue()
for i = 1 to n:
    PQ.enqueue(({c_i}, f_i))
while PQ.size() > 1
    (T_1, f_1) ← PQ.extract_min()
    (T_2, f_2) ← PQ.extract_min()
    PQ.enqueue(({T_1, T_2}, f_1 + f_2))
return PQ.extract_min()
```

$O(n \log n)$

Задача о покрытии множествами
Set Cover



U — множество эл-ов $|U| = N$

$S_1 \dots S_n$ — подмножества U

найти $I \subseteq [1..n]$: $\bigcup\limits_{i \in I} S_i = U$

$|I|$ - min

Алгоритм: на $\forall$ шаге выбираем множество, которое покрывает max кол-во непокрытых эл-ов.

k — опт. размер $|I|$

На 1-м шаге есть множество S_i, кот. покрывает $\geq \frac{N}{k}$ - эл-ов

На j-ом шаге. N_{j-1} - это непокр. эл-ов

$\exists$ множество S', кот. покрывает $\frac{N_{j-1}}{k}$

$$N_j \leq N_{j-1}\left(1 - \frac{1}{k}\right)$$

$N = N_0$

$$N_0\left(1 - \frac{1}{k}\right) \geq N_1$$

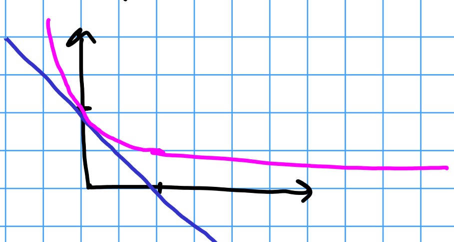
$$N_1\left(1 - \frac{1}{k}\right) \geq N_2$$

$$\dots$$

$$1 \leq N_t \leq N_{t-1}\left(1 - \frac{1}{k}\right) \leq N_{t-2}\left(1 - \frac{1}{k}\right)^2 \leq \dots$$

$$\leq N_0\left(1 - \frac{1}{k}\right)^t \leq N_0 \cdot e^{-t/k}$$

Факт: $1 - x \leq e^{-x}$



$$N_0 \geq e^{t/k}$$

$$\ln N \geq t/k$$

$$t \leq k \cdot \boxed{\ln N}$$