



ITMO UNIVERSITY

# Northern Subregional Contest ACM ICPC 2014–2015, NEERC

## Problem Analysis

IT'sMO *re than a*  
UNIVERSITY



**acm** International Collegiate  
Programming Contest



event  
sponsor



WORLDQUANT.



---

# Жюри соревнований

# Георгий Корнеев



# Андрей Станкевич



# Андрей Лопатин



# Дмитрий Штукенберг



# Павел Маврин



# Максим Буздалов





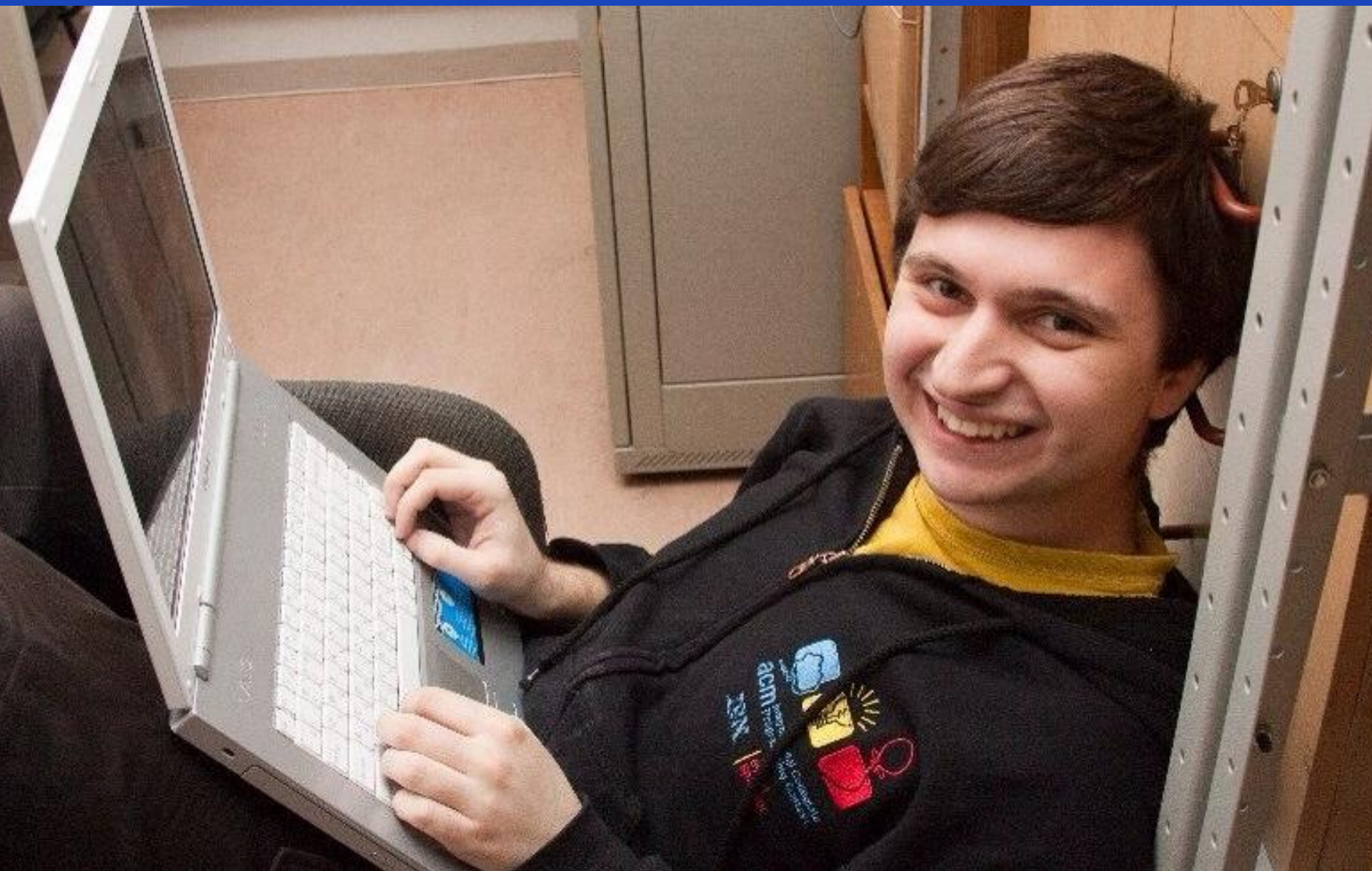
# Сергей Поромов



# Виталий Аксенов



# Михаил Дворкин



# Олег Давыдов



# Юрий Петров



# Дмитрий Гозман



# Егор Куликов



# Нияз Нигматуллин





# Problem A

---

## Alarm Clock

Авторы: Дмитрий Штукенберг

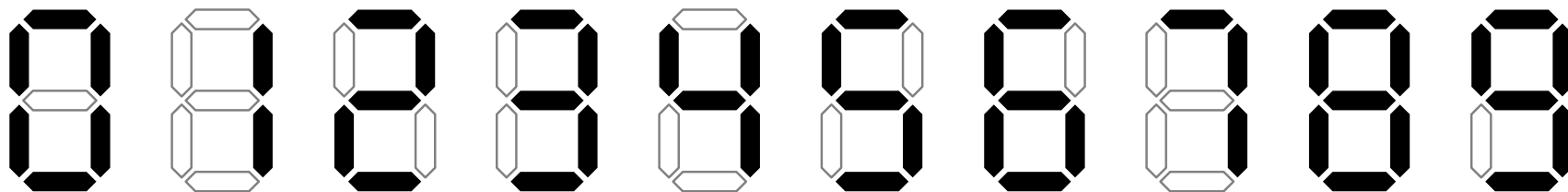
Михаил Дворкин

Условие: Георгий Корнеев

Тесты: Георгий Корнеев

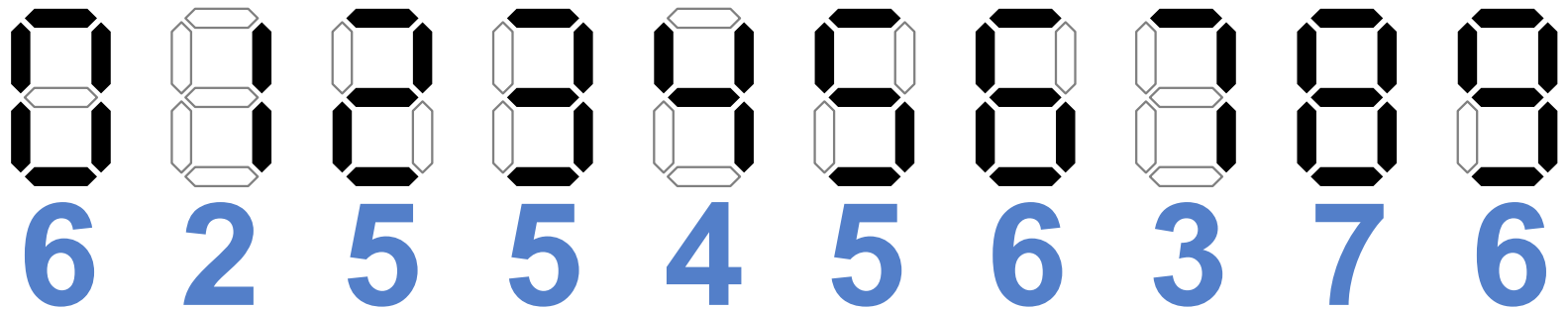
# Постановка задачи

- Дано число сегментов на часах
- Требуется найти время



# Решение

- Посчитать число сегментов



- Перебрать времена hh:mm
  - hh – от 0 до 23
  - mm – от 0 до 59

# Problem B

---

## Buffcraft

Автор: Георгий Корнеев  
Условие: Георгий Корнеев  
Тесты: Георгий Корнеев

# Постановка задачи

- Даны улучшения
  - Непосредственные
  - Процентные

$$(b + d_1 + d_2 + \dots + d_n) \times \\ \times (100 + p_1 + p_2 + \dots + p_m) / 100$$

- Требуется получить максимум при  $n + m \leq k$

# Решение

- Пусть даны  $n$  и  $m$ 
  - Взять наибольшие  $d_i$  и  $p_i$
- Отсортировать  $d_i$  и  $p_i$  по убыванию
  - Посчитать суммы на префиксах
- Перебрать  $n + m = k$ 
  - Вывести максимум
- Время работы –  $O(n \log n)$

# Problem C

---

## Combinator Expression

Автор: Дмитрий Штукенберг  
Условие: Дмитрий Штукенберг  
Тесты: Георгий Корнеев

# Постановка задачи

- Дана комбинаторная формула
  - $Bxyz \Rightarrow x(yz)$
  - $Cxyz \Rightarrow (xz)y$
  - $Kxy \Rightarrow x$
  - $Ix \Rightarrow x$
- Требуется посчитать за минимальное число редукций



# Решение

- Без повторного вычисления
  - Ленивость
  - Подвыражения не копируются
- Не вычислять  $u$  в  $Kxu$

# Неэффективная реализация

- Построить дерево разбора
- Найти самый левый комбинатор
  - Имеющий все аргументы
  - Редуцировать
- Время работы  $O(n^2)$ 
  - $O(n)$  раз по  $O(n)$  действий

# Эффективная реализация

- Рекурсивно вычислять левую ветку
  - Есть аргументы  $\Rightarrow$  редуцировать
  - Нет аргументов  $\Rightarrow$  откат
- Рекурсивно вычислить аргументы
  
- Время работы  $O(n)$ 
  - По одному разу в каждый узел

# Problem D

---

## Digits

Автор: Егор Куликов

Условие: Егор Куликов

Тесты: Егор Куликов

# Постановка задачи

- Дано  $n$
- Требуется
  - Найти  $n$  чисел с одинаковой суммой цифр
  - Сумма чисел должна быть минимальной

# Решение

- Найти  $n$  **минимальных** чисел с одинаковой суммой цифр
- Перебор чисел по возрастанию
  - Подсчет суммы цифр
  - Какая первой встретится  $n$  раз
- Достаточно рассмотреть числа до  $10^6$

# Problem E

---

## Expression

Автор: Георгий Корнеев

Условие: Олег Давыдов

Юрий Петров

Тесты: Олег Давыдов

Юрий Петров

# Постановка задачи

- Даны
  - Регулярное выражение  $E$
  - Подстрока  $S$
- Требуется
  - Найти кратчайшую строку
  - Удовлетворяющую  $E$
  - Содержащую  $S$



# Неэффективное решение

- Ответ  $.^* S.^*$
- Построим НКА для  $E$  и  $.^* S.^*$ 
  - $O(l)$  состояний и переходов
- Пересечем автоматы
  
- Время работы  $O(n^2)$
- Объем памяти  $O(n^2)$

# Входящие и исходящие строки

- Построим НКА для  $E$ 
  - $O(l)$  состояний и переходов
- Для каждого состояния  $T$  найдем
  - Кратчайшую строку, приводящую в него  $in_T$
  - Кратчайшую строку, выводящую из него  $out_T$

# Динамическое программирование

- Перебираем префиксы  $S$ 
  - Для каждого состояния  $in_{T,i}$  кратчайшая строка, заканчивающаяся на  $S[1..i]$
- Переход
  - Переход из состояния по  $S[i]$
  - Замыкание  $\varepsilon$ -переходов в порядке увеличения  $|in_{T,i}|$

# Результат

- Ответ:  $\min_T \{|in_{T,|S|}| + |out_T|\}$
- Время работы  $O(|S|l)$ 
  - $in_T, out_T$  – обход в ширину,  $O(l)$
  - $in_{T,i}$  –  $|S|$  наращиваний и обходов в глубину,  $O(|S|l)$
- Объем памяти  $O(l)$

# Problem F

---

## Fragmentation

Автор: Георгий Корнеев

Условие: Павел Маврин

Тесты: Павел Маврин

# Постановка задачи

- Дана последовательность  $\{a_i\}$
- Требуется разрезать ее на минимальное число частей, переставив которые можно отсортировать  $\{a_i\}$

# Основные идеи

- Объединение равных соседей
- Экономия разреза  $a_i + 1 = a_{i+1}$ 
  - Возможна не всегда

# Динамическое программирование

- По возрастанию  $a_i$ 
  - Экономия после  $j$
  - Существует один кусок  $a_i = j$ 
$$E_{j,i+1} = E_{j-1,i} + 1$$
  - Существует несколько кусков
$$E_{j,i+1} = \max_{k \neq i} \{E_{j-1,k}\} + 1$$
  - Храним два лучших результата



# Время работы

- $O(n \log n)$ 
  - Сортировка  $a_i$  –  $O(n \log n)$
  - Динамическое программирование –  $O(n)$  в сумме

# Problem G

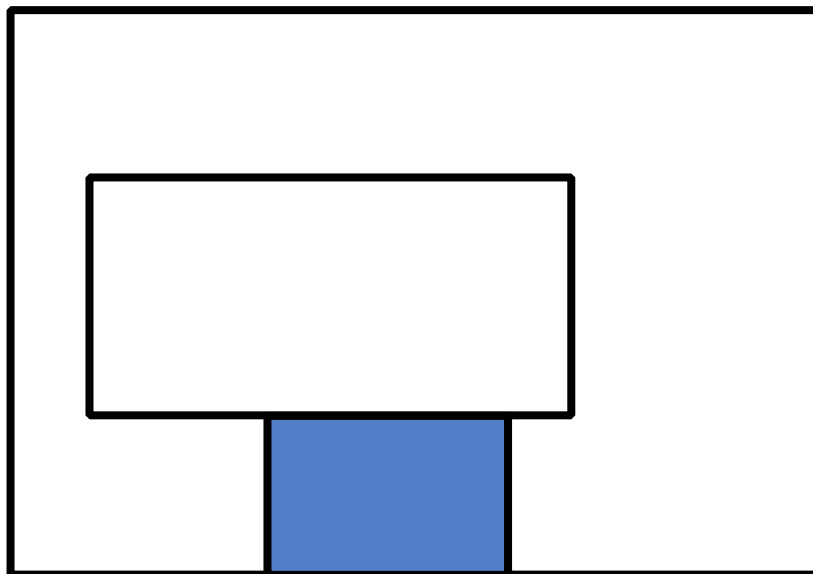
---

## Grave

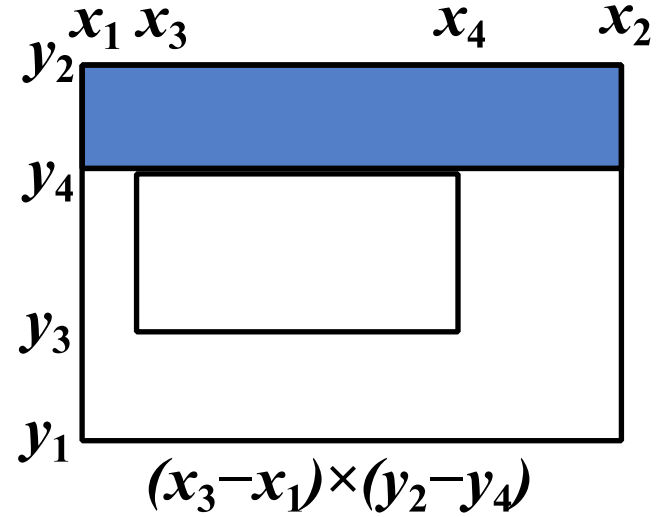
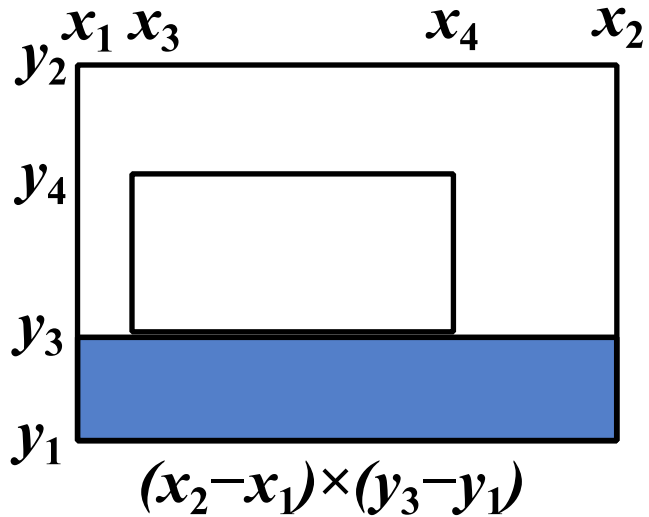
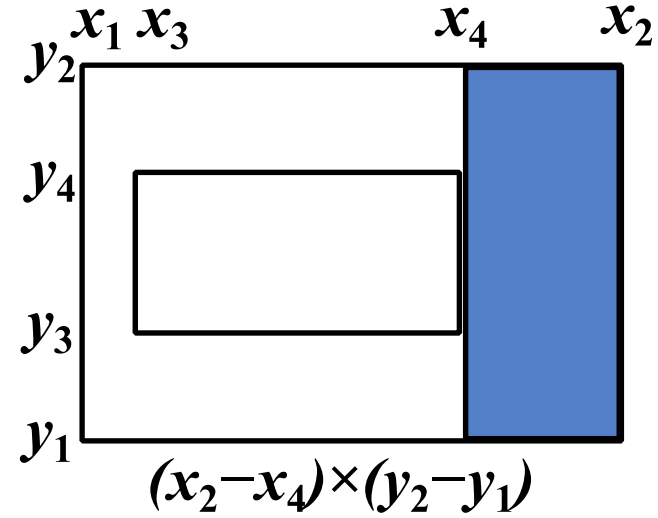
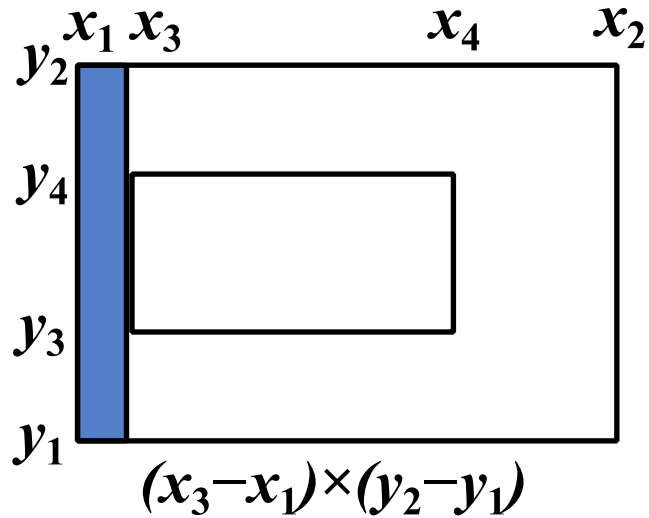
Автор: Виталий Аксенов  
Условие: Сергей Поромов  
Тесты: Сергей Поромов

# Постановка задачи

- Дан прямоугольник с прямоугольной дырой
- Требуется разместить прямоугольник  $w \times h$



# Максимальные прямоугольники



# Решение

- Новый прямоугольник должен помещаться в одном из максимальных
- Время работы  $O(1)$

# Problem H

---

## Hiking in the Hills

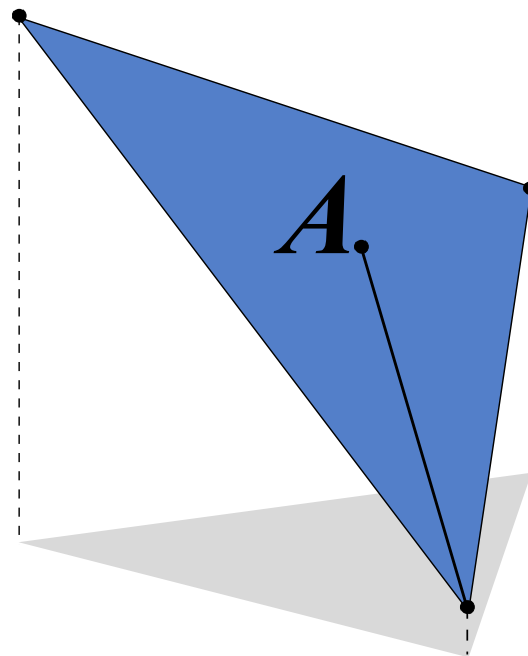
Автор: Георгий Корнеев  
Условие: Михаил Дворкин  
Тесты: Михаил Дворкин

# Постановка задачи

- Дан ландшафт из треугольников
- Требуется найти путь от старта до финиша с минимальной максимальной высотой

# Старт и финиш

- Найдем стартовый треугольник
  - Минимальная по высоте вершина
  - Соединим ребром со стартом
- Аналогично для финиша

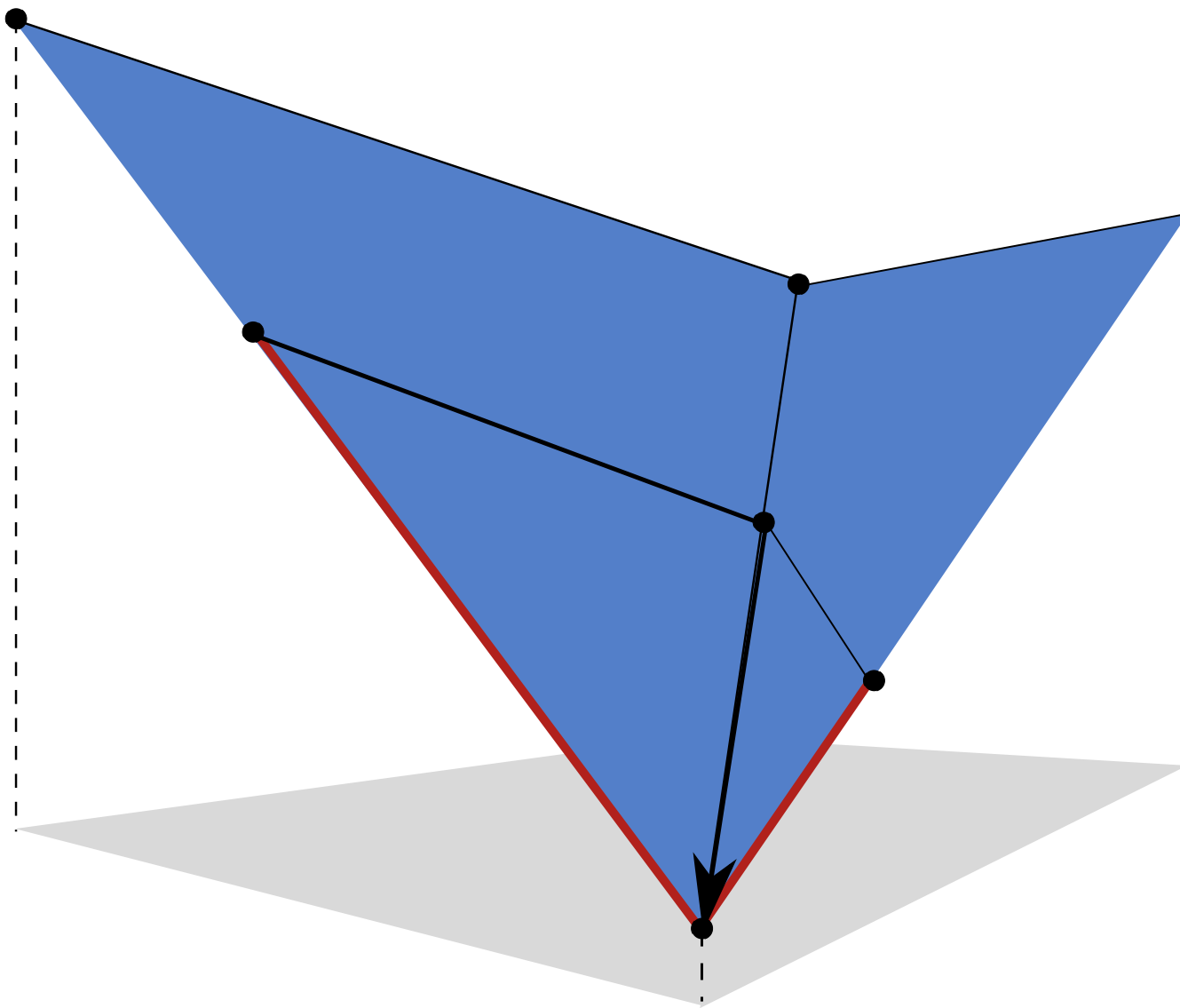




# Основная идея

- Достаточно ходить по ребрам
  - Вход в треугольник – на ребре
  - Выход – на ребре
  - Существует путь от входа до выхода с  $h \leq \max(h_{\text{входа}}, h_{\text{выхода}})$

# Основная идея



# Решение

- Модифицированный алгоритм Прима
  - Вес – максимальная высота
  
- Время работы  $O(n \log n)$ 
  - $E = O(n)$
  - Допустимо  $O(n^2)$

# Problem I

---

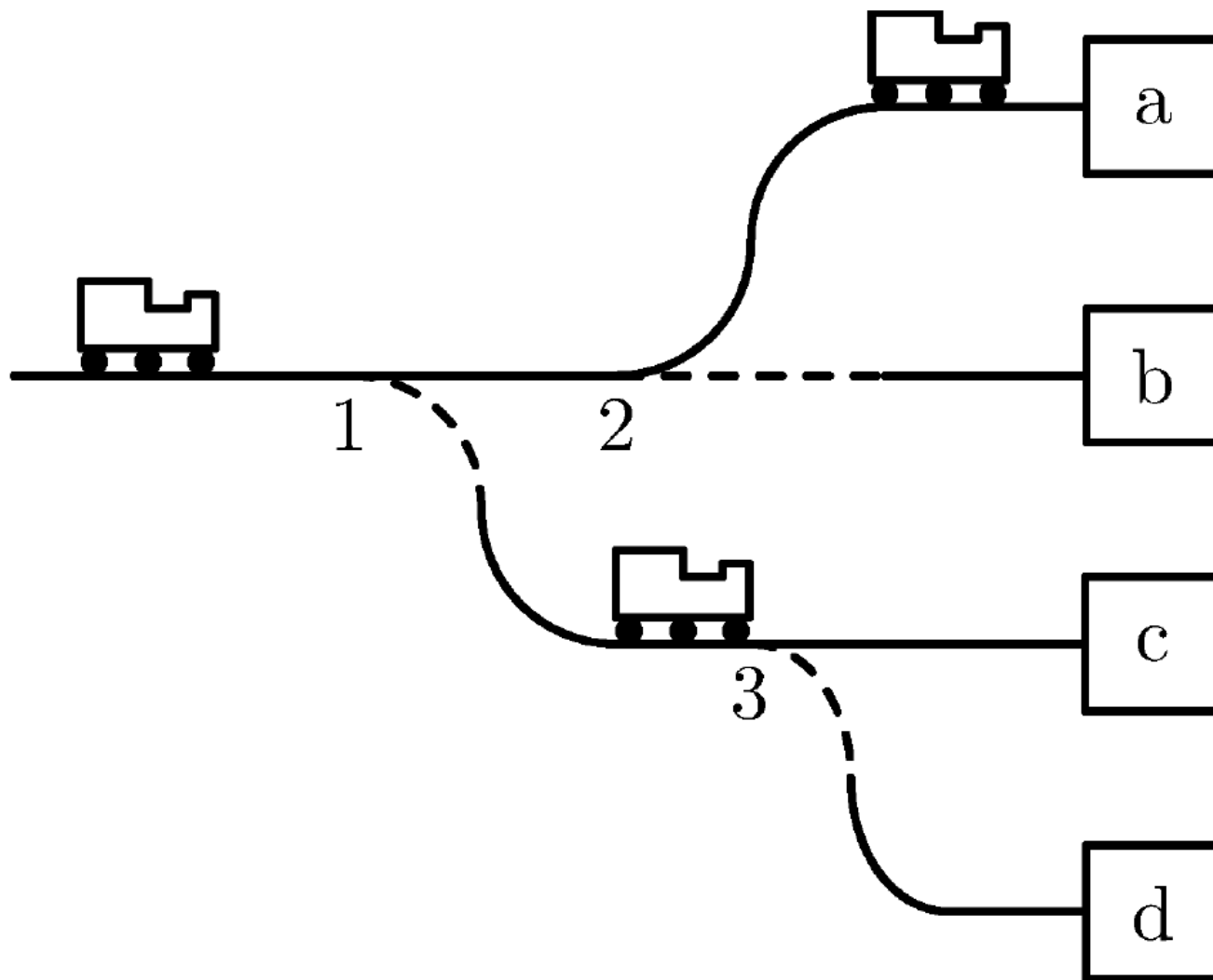
## Instruction

Автор: Дмитрий Гозман  
Условие: Дмитрий Гозман  
Тесты: Дмитрий Гозман

# Постановка задачи

- Даны
  - Схема станции
  - Расписание поездов
- Требуется составить расписание переключения стрелок

# Постановка задачи



# Решение

- Схема станции – дерево
  - Поезда въезжают в разное время  $\Rightarrow$  проезжают стрелку в разное время
  - Переключаем стрелку перед поездом
  
- Время работы –  $O(nm)$

# Problem J

---

## Joy of Flight

Автор: Георгий Корнеев

Условие: Нияз Нигматуллин

Тесты: Нияз Нигматуллин



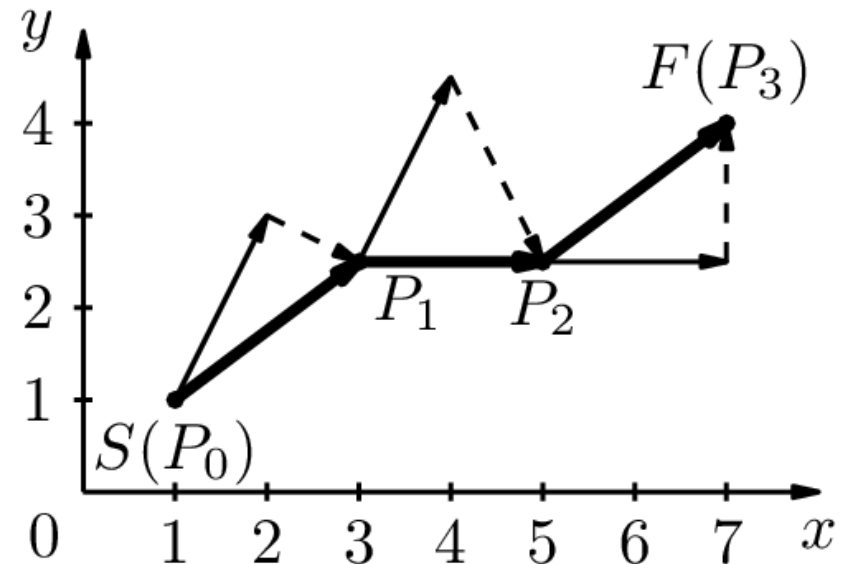
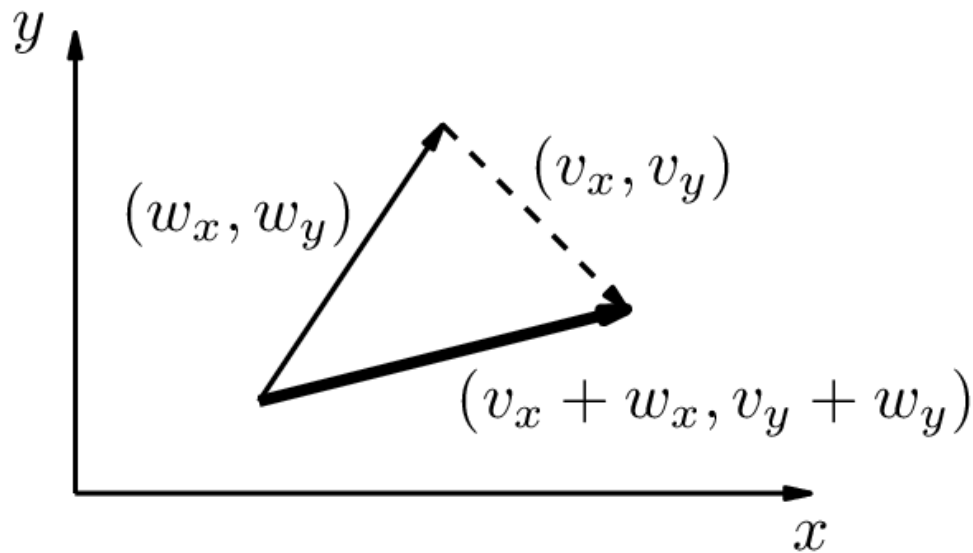
# Постановка задачи

- Даны
  - Самолет со скоростью  $v_{max}$
  - Изменяющийся ветер  $(w_{x_i}, w_{y_i})$
- Требуется долететь из  $S$  в  $T$  за время  $k$

# Постановка задачи

- $\sqrt{w_{x_i}^2 + w_{y_i}^2} \leq v_{max}$

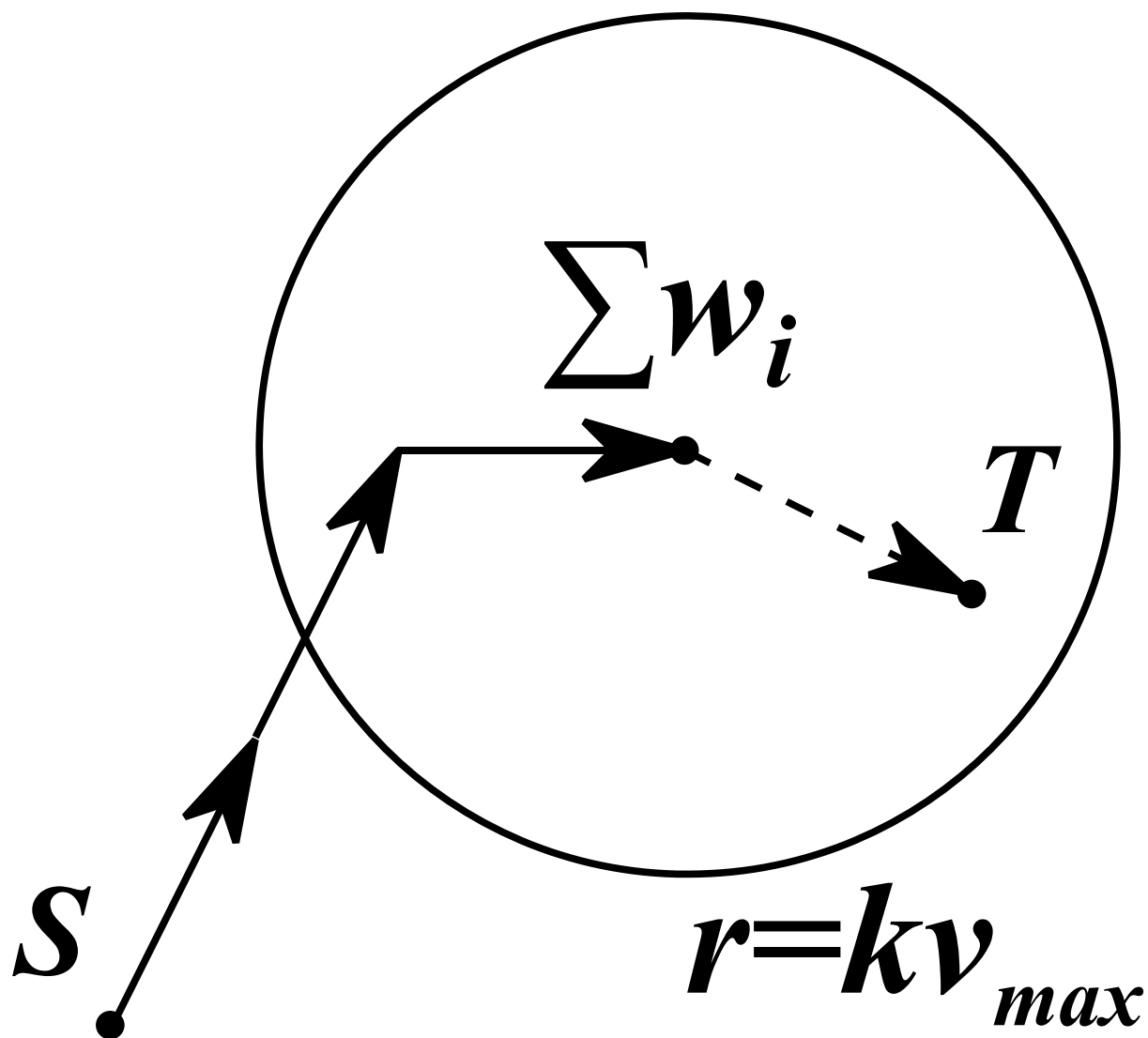
- Можем стоять на месте



# Основная идея

- Подсчитаем, куда снесет из  $S$  за все время
- Построим круг радиуса  $kv_{max}$
- Если  $F$  в круге – то можно
  - Летим по прямой

# Основная идея



# Решение

- С какой точностью можно определить, можно ли долететь?
  - С абсолютной – сравнивать квадраты величин
- Как лететь
  - Равномерно по прямой
- Время работы  $O(n)$

# Problem K

---

## **Kebab House**

Автор: Виталий Аксенов  
Условие: Виталий Аксенов  
Тесты: Виталий Аксенов

# Постановка задачи

- Даны
  - Задания  $(q_i, x_i)$  – размер, качество
  - Можно пропускать ингредиенты не чаще одного в  $t$  секунд
- Требуется посчитать число сценариев, удовлетворяющих ограничениям на качество

# Динамическое программирование

- $C_{i,j,k}$  – число способов положить  $j$  из первых  $i$  ингредиентов так, что бы с последнего пропуска прошло  $k$ 
  - $C_{i,j,k} = C_{i-1,j-1,k+1}$  , при  $k \neq 0$
  - $C_{i,j,0} = C_{i-1,j-1,0} + C_{i-1,j,t}$  , при  $k = 0$
- **Время работы –  $O(nq^2t)$**



# Оптимизация (вариант 1)

- Замена координат
  - $j' = j - i, k' = k + i$
  - $C_{i,j',k'} = C_{i-1,j',k'}$ , при  $k' \neq i$
  - $C_{i,j',i} = C_{i-1,j',i} + C_{i-1,j'-1,i+t}$ , при  $k' = i$
  - При  $k' \neq i$  можно ничего не делать
- Время работы –  $O(nqt)$

# Оптимизация (вариант 2)

- Нельзя пропустить больше чем  $\left\lceil \frac{q_i}{t+1} \right\rceil$  ингредиентов
- Заменяем  $j' = i - j$ 
  - Достаточно считать для  $j' \leq \left\lceil \frac{q_i}{t+1} \right\rceil$
- Время работы  $O(nq^2) = O\left(nq \left\lceil \frac{q}{t+1} \right\rceil t\right)$

# Альтернативное решение (1)

- Предподсчитаем  $C_{i,j,k}$  для  $i \leq q_{max}$ , при условии, что в начале можно сразу пропустить ингредиент
- $S_{i,j,k} = \sum_{x=j}^i C_{i,x,k}$  – не менее  $j$  ингредиентов

# Альтернативное решение (2)

- $A_{p,k}$  – выполнен  $p$ -й заказ и прошло  $k$  секунд с предыдущего пропуска
  - $A_{p,k} = \sum_{j=0}^t A_{p-1,j} S_{q_p-(t-j), x_{p,k}}$
- Ответ –  $\sum_{k=0}^t A_{n,k}$
- Время работы –  $O(nt^2 + q^2t)$

# Вопросы