ITMO UNIVERSITY

Northern Subregional Contest ACM ICPC 2015–2016, NEERC

Problem Analysis

ITMOre than a UNIVERSITY

acm International Collegiate Programming Contest

IBM event sponsor

DEVEXPERTS

@mail.ru group

JetBRAINS

WORLDQUANT

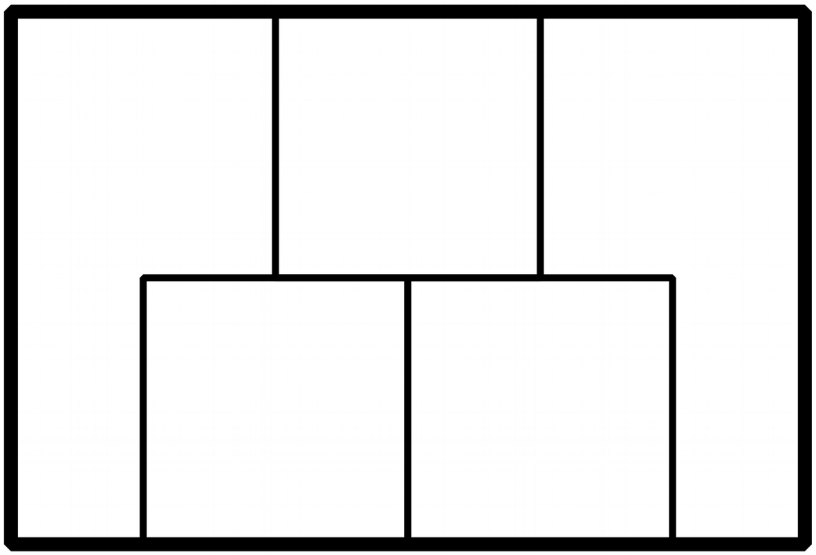
Yandex

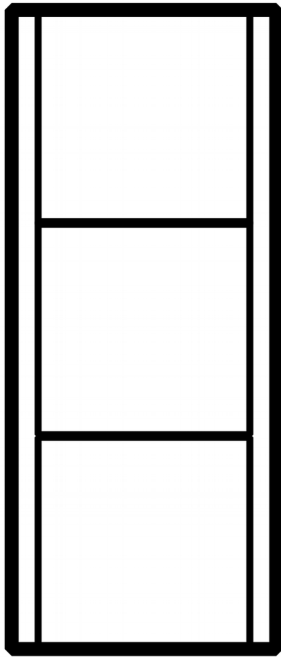
Problem A

Alex Origami Squares

Problem statement

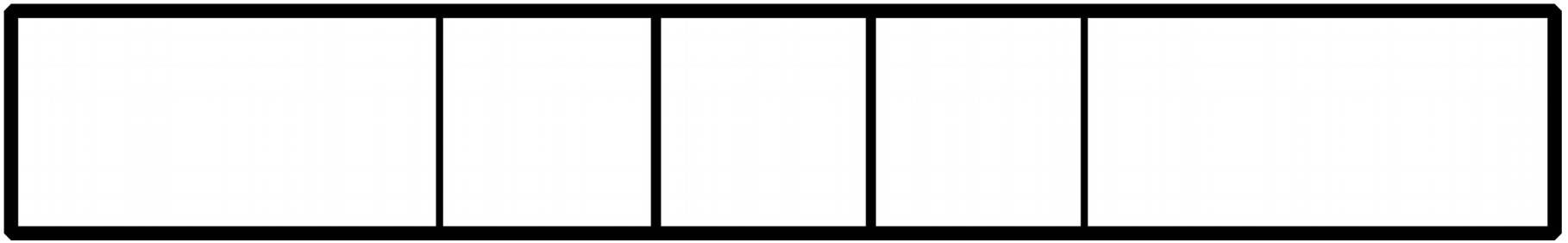
- Given a rectangle h by w
- Cut 3 equal maximum squares





Solution

```
max(
    min / 2.0,      // Case 1
    min(
        max / 3.0,// Case 2
        min,      // Case 3
    ))
```

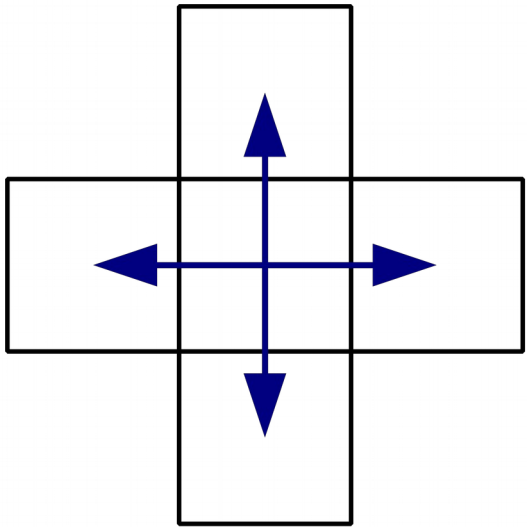


Problem B

Black and White

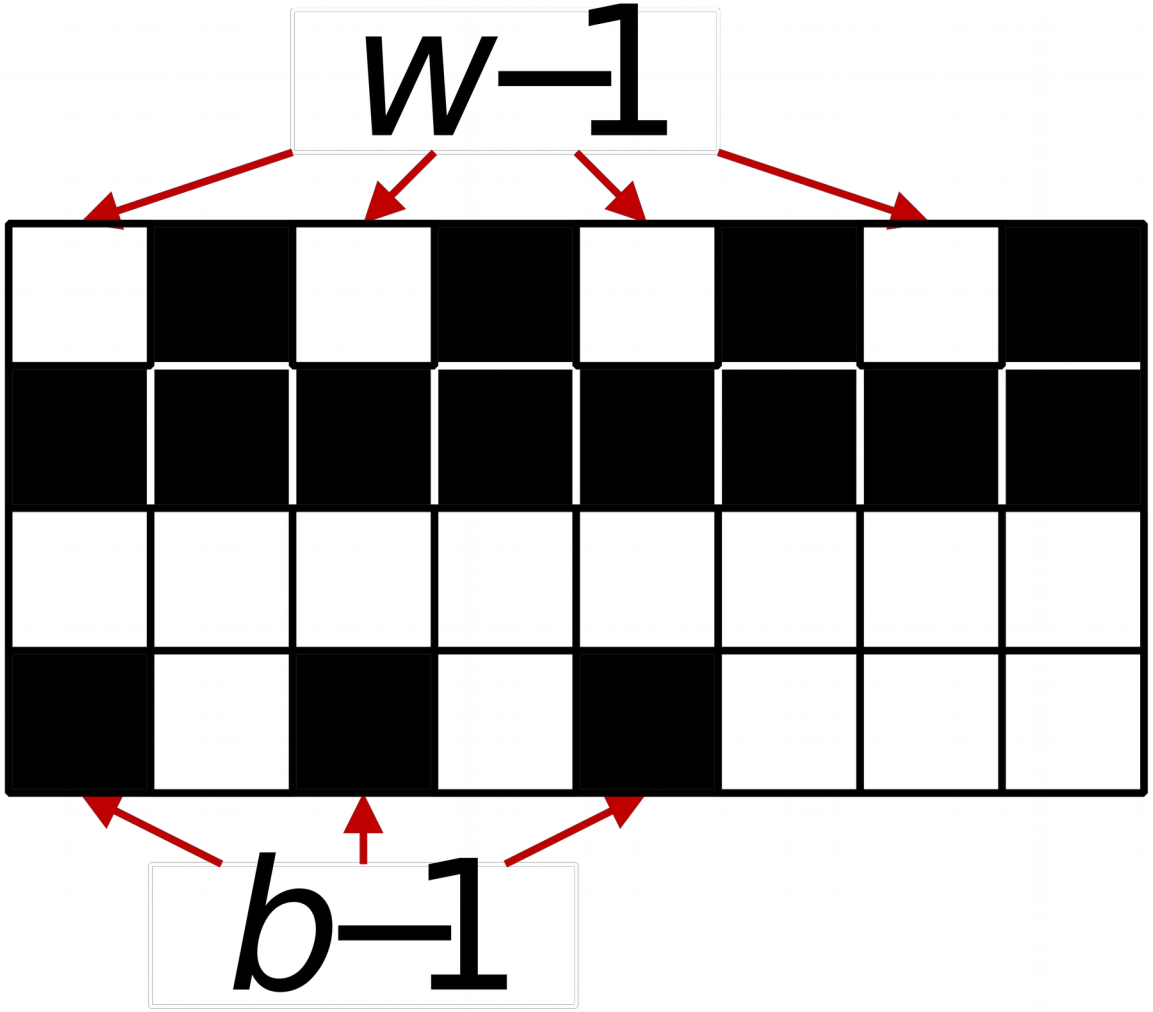
Problem statement

- Build a black and white grid
- B black regions черных областей
- W white regions
- Regions are 4-connected



Problem analysis

- This construction works



Problem C

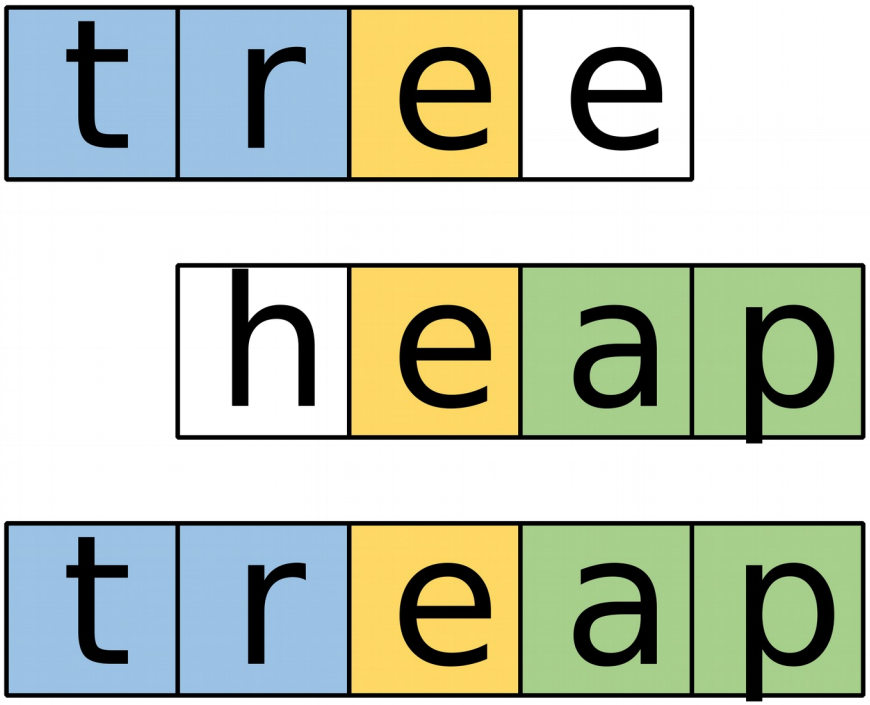
Concatenation

Problem statement

- Calculate the number of strings
  - Formed by a concatenation of
  - Nonempty prefix of S1
  - Nonempty suffix of S2

Solution idea

- When duplicates happen?
  - Same letter in 2 strings



Solution

- Calculate the frequencies
  - C_{1i} – frequency of i in S1
  - C_{2i} – frequency of i in S2
- Answer is

$$|S_1||S_2| - \sum_{i=a}^{z} C_{1i}C_{2i}$$

- Look out for int overflow

Problem D

Distribution in Metagonia

Problem statement

- Represent N as sum of terms $2^a 3^b$
  - Any number shouldn't be divisible by other

$$10 = 4 + 6$$

Solution

- For even n
  - Solve for n/2
  - Multiply everything by 2
- For odd n
  - Solve for $n - 3^b$, where b is max possible
  - Add 3^b to the result

Problem E

Easy Arithmetic

Problem statement

- Add plus and minus signs to expression
  - Maximize the result

$$10 + 20 - 30$$
$$\Downarrow$$
$$10 + 20 - 3 + 0$$

Solution

- $+d_1d_2...d_k \rightarrow$ leave as is
- $-d_1d_2...d_k \rightarrow$ transform into $-d_1+d_2...d_k$
  - Except for $d_2=0$
  - $-d_1+0+d_3...d_k$

Problem F

Fygon

Автор: Георгий Корнеев
Условие: Павел Маврин
Тесты: Павел Маврин

Problem statement

- Calculate the number of **lag** calls
- Output as polynomial of n
- For-loops

```
for i in range(n):
    for j in range(i):
        for x in range(5):
            lag
lag
```

Solution

- Polynomial's degree
  - Equal to the number of nested loops
  - Less than or equal to 6
- Answer
  - Interpret for n=0,1,…6
  - Use interpolation formulas

Problem G

Graph

Problem statement

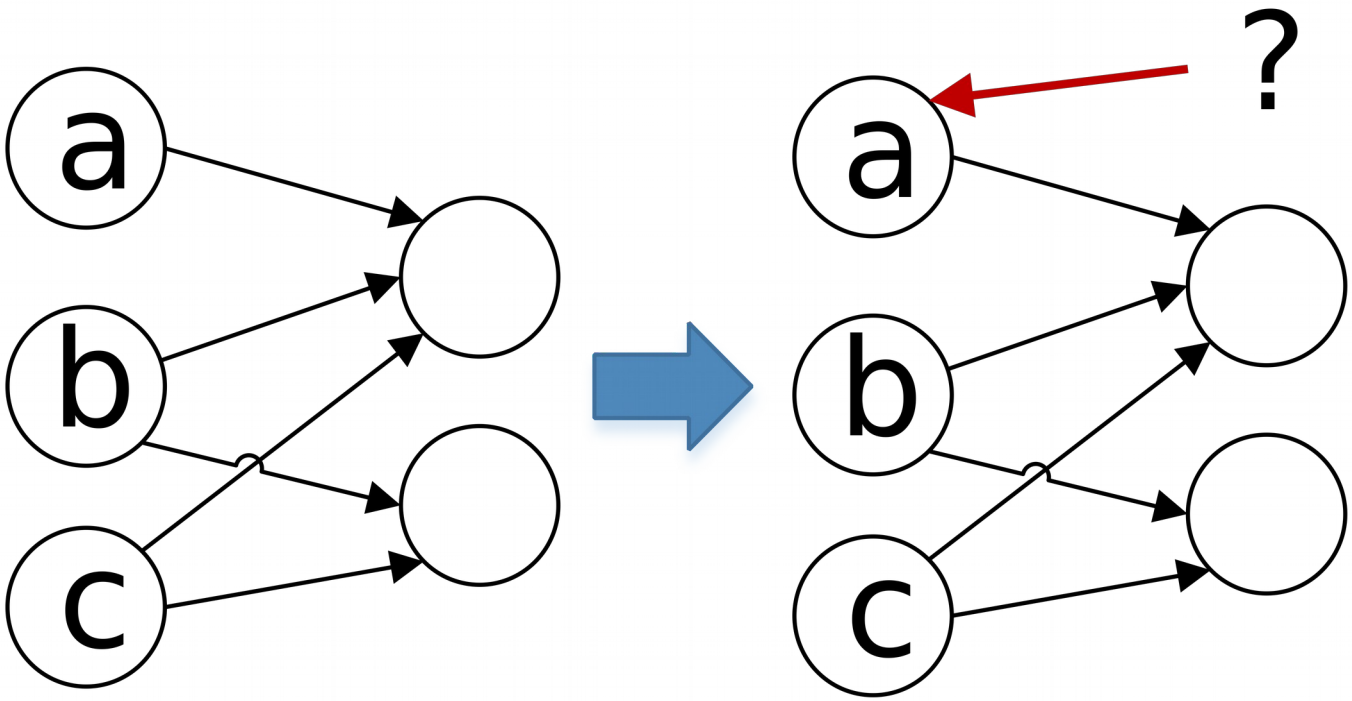
- Given a DAG
- Add at most k edges
  - Maximize minimum possible topological sort

Solution

- Construct a lex-min topological sort
- You can add a vertex when:
  - No incoming edges
  - Choose minimum of those

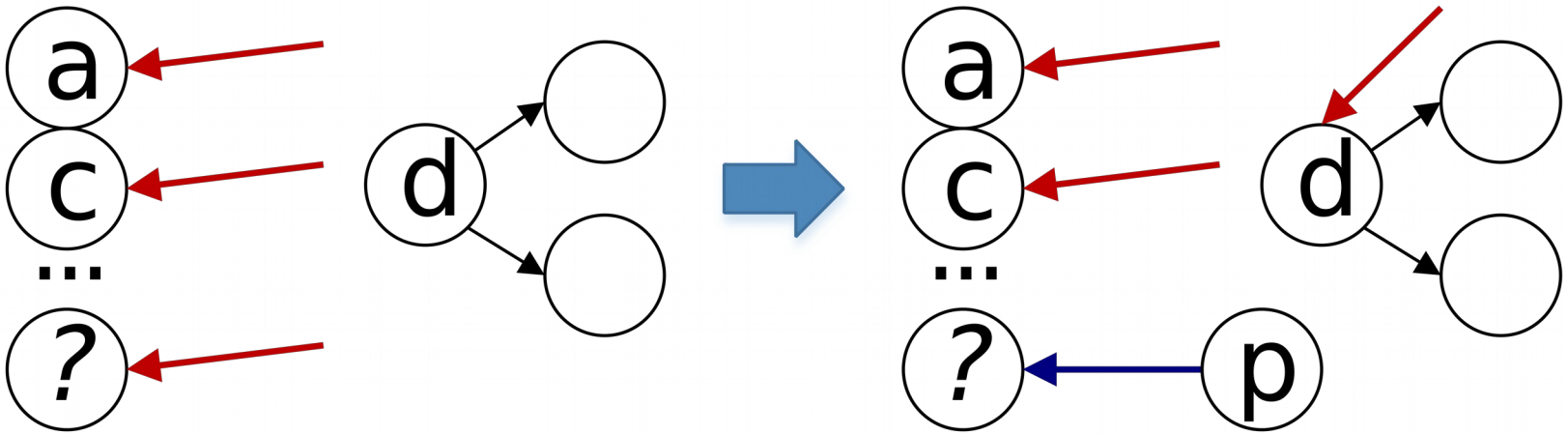
Inserting new edges

- Many possible vertices
  - Add edge leading to minimal vertex



Inserting new edges

- Single candidate remaining
  - ? < d ⇒ print and remove d
  - ? > d ⇒ insert new edge to ? from last vertex in the output



Implementation

- Vertex
  - In/out degree
  - Remove an edge in O(1)
- Ordered set
  - Candidates
  - Vertices with an edge

Problem H

Hash Code Hacker

Problem statement

- Generate n strings with equal hash codes
- Polynomial hash

```
s[0]*31^(n-1) +
s[1]*31^(n-2) +
... +
s[n-1]
```

Solution

- From sample output
- hash(ed) = hash(fe) = h
- Construct 4 strings
- hash(eded) = hash(edfe) = hash(feed) = hash(fefe) = $(31^2 + 1)h$

Problem I

Insider’s Information

Problem statement

- Unknown order
  - M triples $a_i < b_i < c_i$
  - Or $a_i > b_i > c_i$
- Find any arrangement
  - Satisfying at least m/2 triples

Solution

- Triples correspond to some “real” order
  - Some x is never in the middle
  - Save all triples with $a_i = x$ or $c_i = x$
- Throw x away and solve recursively
  - Set x to minimum or maximum of all
  - Saved triples determine which one
  - Choose the majority

Problem J

Journey to the "The World's Start"

Problem statement

- Go from one metro station to another
- Maximum time T
- Re-entering takes some time
- Travel cards allowing to travel r stations
  - Find the cheapest one

Solution

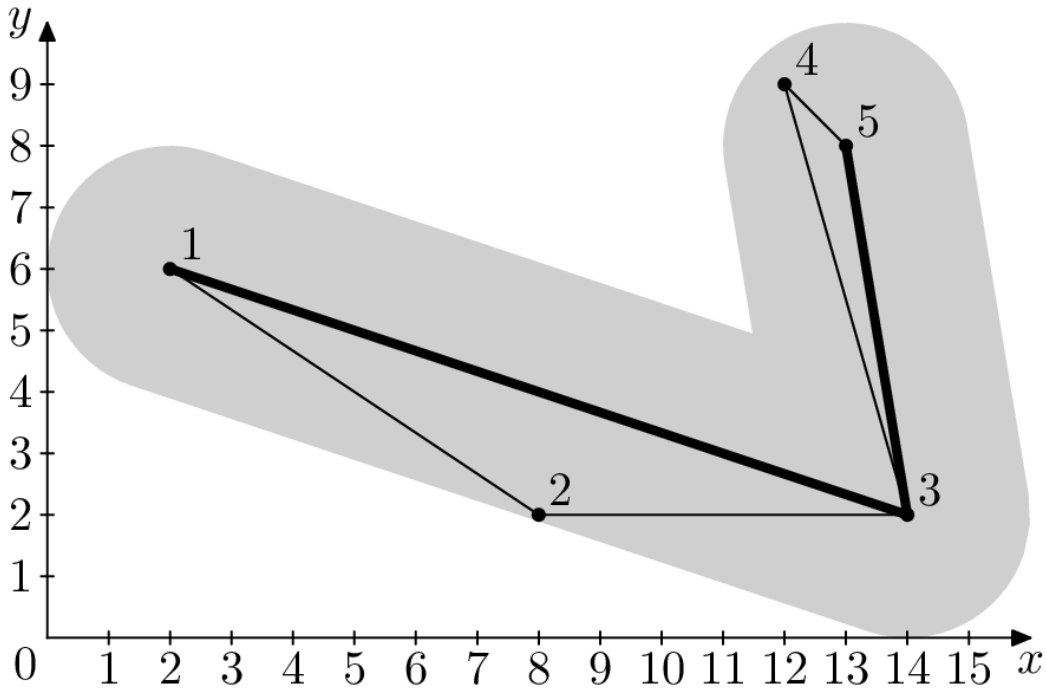
- Binary search on r
- Find the minimum time to reach finish
- Dynamic programming
- $T_i = d_i + \min_{i-r' \le j < i} T_j$
- Can be done in O(n) using queue with minimum
- $O(n \log^2 n)$ is also accepted

Problem K

Kingdom Trip

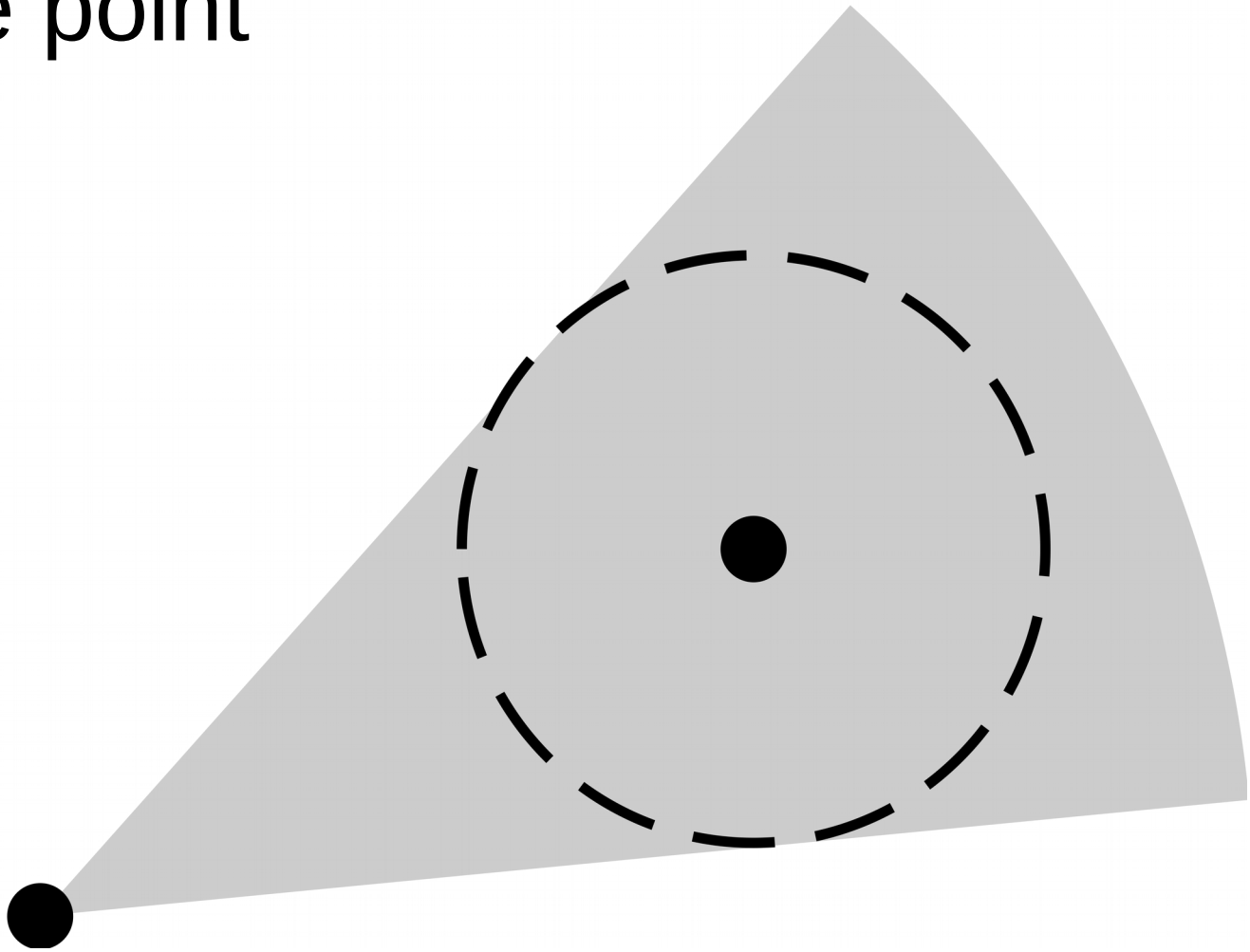
Problem statement

- «Straighten» the path
- Distance from the new path to old vertices



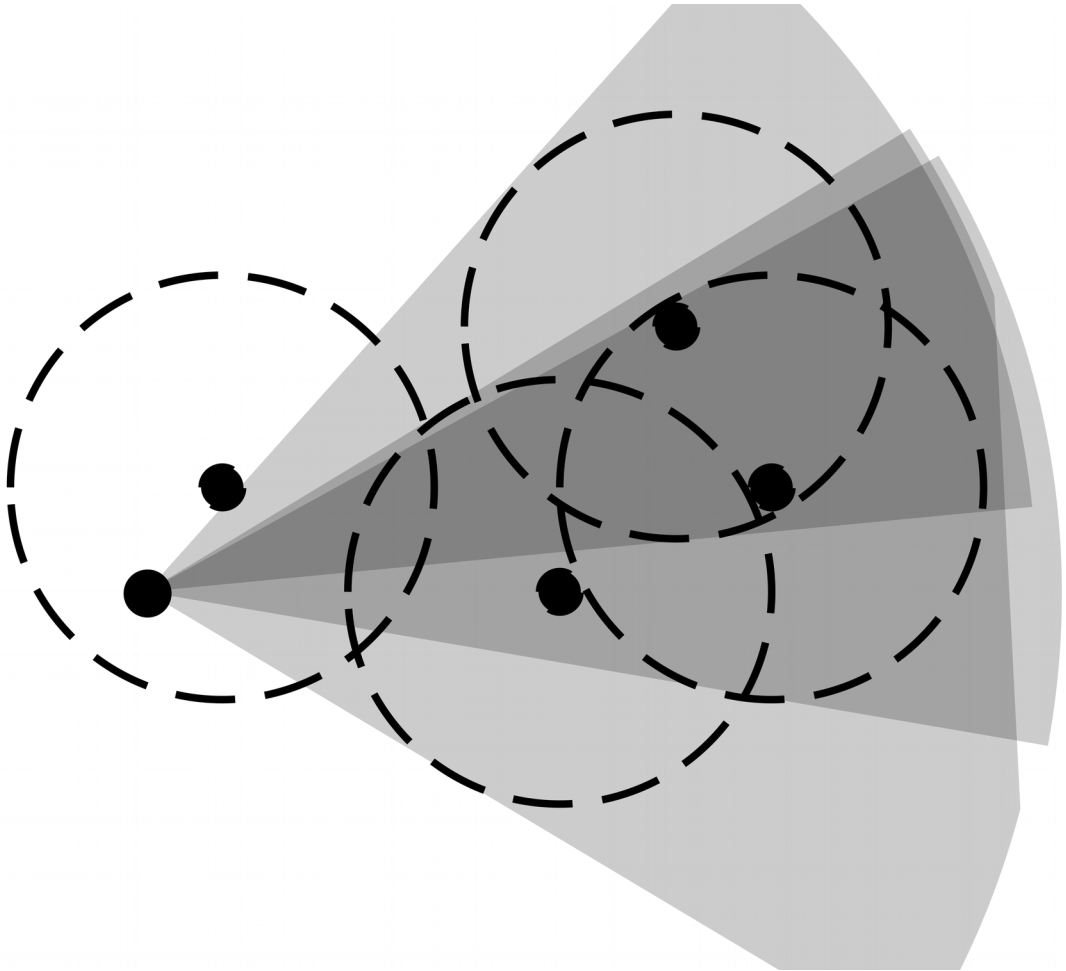
Allowed ray position

- One point



Allowed ray position

- Many points



Solution

- Iterate over all starting points
- Iterate over all ending points
  - Intersect all angles
  - Ending vertex is good if lies in the angle
- After that, do a DP in $O(n^2)$

Problem L

Lucky Chances

Problem statement

- Given a matrix of numbers
- Find
  - For each element
  - For each direction
  - If this element is strictly greater than all numbers in this direction