# Problem Analysis — First Day Contest

Ivan Kazmenko

St. Petersburg State University

Sunday, August 24, 2014

# Additive Class — Preliminaries

*In this problem, you have to find how many elements of the additive class of* $a$ *and* $b$ *belong to* $[l, r]$*.*

# Additive Class — Preliminaries

*In this problem, you have to find how many elements of the additive class of $a$ and $b$ belong to $[l, r]$.*

- Let $d = \mathrm{GCD}(a, b)$.
  $a \leftarrow \frac{a}{d}$, $b \leftarrow \frac{b}{d}$, $l \leftarrow \lceil \frac{l}{d} \rceil$, $r \leftarrow \lfloor \frac{r}{d} \rfloor$.

# Additive Class — Preliminaries

*In this problem, you have to find how many elements of the additive class of a and b belong to $[l, r]$.*

- Let $d = \mathrm{GCD}(a, b)$.
  $a \leftarrow \frac{a}{d}$, $b \leftarrow \frac{b}{d}$, $l \leftarrow \lceil \frac{l}{d} \rceil$, $r \leftarrow \lfloor \frac{r}{d} \rfloor$.
- Now $\mathrm{GCD}(a, b) = 1$.
  Every $u \geqslant a \cdot b$ belongs to the class.
  Every $v < a \cdot b$ has 0 or 1 representation as $x \cdot a + y \cdot b$.

# Additive Class — Preliminaries

*In this problem, you have to find how many elements of the additive class of a and b belong to $[l, r]$.*

- Let $d = \mathrm{GCD}(a, b)$.
  $a \leftarrow \frac{a}{d}$, $b \leftarrow \frac{b}{d}$, $l \leftarrow \lceil \frac{l}{d} \rceil$, $r \leftarrow \lfloor \frac{r}{d} \rfloor$.
- Now $\mathrm{GCD}(a, b) = 1$.
  Every $u \geqslant a \cdot b$ belongs to the class.
  Every $v < a \cdot b$ has 0 or 1 representation as $x \cdot a + y \cdot b$.
- $f(l, r) = g(r) - g(l - 1)$.

# Additive Class — Illustration

Example: $a = 17$, $b = 10$.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | **sum** | **diff** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | x | . | . | . | . | . | . | . | . | . | x | . | . | . | . | . | . | 2 | 2 |
| 17 | x | . | . | x | . | . | . | . | . | . | x | . | . | x | . | . | . | 4 | 2 |
| 34 | x | . | . | x | . | . | x | . | . | . | x | . | . | x | . | . | x | 6 | 2 |
| 51 | x | . | . | x | . | . | x | . | . | x | x | . | . | x | . | . | x | 7 | 1 |
| 68 | x | . | x | x | . | . | x | . | . | x | x | . | x | x | . | . | x | 9 | 2 |
| 85 | x | . | x | x | . | x | x | . | . | x | x | . | x | x | . | x | x | 11 | 2 |
| 102 | x | . | x | x | . | x | x | . | x | x | x | . | x | x | . | x | x | 12 | 1 |
| 119 | x | x | x | x | . | x | x | . | x | x | x | x | x | x | . | x | x | 14 | 2 |
| 136 | x | x | x | x | x | x | x | . | x | x | x | x | x | x | x | x | x | 16 | 2 |
| 153 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | 17 | 1 |

# Additive Class — Illustration

Example: $a = 17$, $b = 10$.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | **sum** | **diff** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | x | . | . | . | . | . | . | . | . | . | x | . | . | . | . | . | . | 2 | 2 |
| 17 | x | . | . | x | . | . | . | . | . | . | x | . | . | x | . | . | . | 4 | 2 |
| 34 | x | . | . | x | . | . | x | . | . | . | x | . | . | x | . | . | x | 6 | 2 |
| 51 | x | . | . | x | . | . | x | . | . | x | x | . | . | x | . | . | x | 7 | 1 |
| 68 | x | . | x | x | . | . | x | . | . | x | x | . | x | x | . | . | x | 9 | 2 |
| 85 | x | . | x | x | . | x | x | . | . | x | x | . | x | x | . | x | x | 11 | 2 |
| 102 | x | . | x | x | . | x | x | . | x | x | x | . | x | x | . | x | x | 12 | 1 |
| 119 | x | x | x | x | . | x | x | . | x | x | x | x | x | x | . | x | x | 14 | 2 |
| 136 | x | x | x | x | x | x | x | . | x | x | x | x | x | x | x | x | x | 16 | 2 |
| 153 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | 17 | 1 |

The **diff** column: Euclid's algorithm (continued fraction).

$$\frac{17}{10} = 1 + \frac{1}{\frac{10}{7}} = 1 + \frac{1}{1 + \frac{1}{\frac{7}{3}}} = 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{3}}} = [1; 1, 2, 3]$$

# Additive Class — Illustration

Last line: shift the table.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | sum | diff |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | . | . | . | . | . | . | . | . | . | x | . | . | . | . | . | . | . | 1 | 1 |
| 17 | . | . | x | . | . | . | . | . | . | x | . | . | x | . | . | . | . | 3 | 2 |
| 34 | . | . | x | . | . | x | . | . | . | x | . | . | x | . | . | x | . | 5 | 2 |
| 51 | . | . | x | . | . | x | . | . | x | x | . | . | x | . | . | x | . | 6 | 1 |
| 68 | . | x | x | . | . | x | . | . | x | x | . | x | x | . | . | x | . | 8 | 2 |
| 85 | . | x | x | . | x | x | . | . | x | x | . | x | x | . | x | x | . | 10 | 2 |
| 102 | . | x | x | . | x | x | . | x | x | x | . | x | x | . | x | x | . | 11 | 1 |
| 119 | x | x | x | . | x | x | . | x | x | x | x | x | x | . | x | x | . | 13 | 2 |
| 136 | x | x | x | x | x | x | . | x | x | x | x | x | x | x | x | x | . | 15 | 2 |
| 153 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | 17 | 2 |

# Additive Class — Illustration

Last line: shift the table.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | **sum** | **diff** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | . | . | . | . | . | . | . | . | . | x | . | . | . | . | . | . | . | 1 | 1 |
| 17 | . | . | x | . | . | . | . | . | . | x | . | . | x | . | . | . | . | 3 | 2 |
| 34 | . | . | x | . | . | x | . | . | . | x | . | . | x | . | . | x | . | 5 | 2 |
| 51 | . | . | x | . | . | x | . | . | x | x | . | . | x | . | . | x | . | 6 | 1 |
| 68 | . | x | x | . | . | x | . | . | x | x | . | x | x | . | . | x | . | 8 | 2 |
| 85 | . | x | x | . | x | x | . | . | x | x | . | x | x | . | x | x | . | 10 | 2 |
| 102 | . | x | x | . | x | x | . | x | x | x | . | x | x | . | x | x | . | 11 | 1 |
| 119 | x | x | x | . | x | x | . | x | x | x | x | x | x | . | x | x | . | 13 | 2 |
| 136 | x | x | x | x | x | x | . | x | x | x | x | x | x | x | x | x | . | 15 | 2 |
| 153 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | 17 | 2 |

The **diff** column is cyclically shifted by $(a^{-1} \bmod b)$ upwards.

# Board Blitz — Solution 1

*In this problem, you have to quickly find the final position in a game on a square board.*

# Board Blitz — Solution 1

*In this problem, you have to quickly find the final position in a game on a square board.*

- Pack the board into bits: `int64`.

# Board Blitz — Solution 1

*In this problem, you have to quickly find the final position in a game on a square board.*

- Pack the board into bits: `int64`.
- Movement: bit shift ($+8$, $+1$, $-8$, $-1$).

# Board Blitz — Solution 1

*In this problem, you have to quickly find the final position in a game on a square board.*

- Pack the board into bits: `int64`.
- Movement: bit shift ($+8$, $+1$, $-8$, $-1$).
- Rotation: 1. Naive $O(\mathrm{bits})$.

# Board Blitz — Solution 1

*In this problem, you have to quickly find the final position in a game on a square board.*

- Pack the board into bits: `int64`.
- Movement: bit shift ($+8$, $+1$, $-8$, $-1$).
- Rotation: 1. Naive $O(\text{bits})$.
- Rotation: 2. Clever $O(\log(\text{bits}))$.

# Board Blitz — Solution 1

*In this problem, you have to quickly find the final position in a game on a square board.*

- Pack the board into bits: `int64`.
- Movement: bit shift $(+8, +1, -8, -1)$.
- Rotation: 1. Naive $O(\mathrm{bits})$.
- Rotation: 2. Clever $O(\log(\mathrm{bits}))$.
- Rotation: 3. Blunt $O(\mathrm{bits}/K)$ with $2^K$ precalculation.

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

- Estimate the lifetime of a particular game piece.

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

- Estimate the lifetime of a particular game piece.
- 1. Pretend: only one player, only moving, only `right`.
  Off the board after 8 turns.

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

- Estimate the lifetime of a particular game piece.
- 1. Pretend: only one player, only moving, only `right`.
  Off the board after 8 turns.
- 2. Pretend: only one player, only moving, only `left` and `right`.
  Probability of $8 \times \texttt{right}$: $1/2^8$.

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

- Estimate the lifetime of a particular game piece.
- ...
- 2. Pretend: only one player, only moving, only `left` and `right`.
  Probability of $8 \times \texttt{right}$: $1/2^8$.
- 3. Pretend: only one player, only moving.
  Probability of $8 \times \texttt{right}$: $1/2^8$, time $\times 2$.

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

- Estimate the lifetime of a particular game piece.
- ...
- 3. Pretend: only one player, only moving.
  Probability of $8 \times \texttt{right}$: $1/2^8$, time $\times 2$.
- 4. Pretend: only one player, allow rotation.
  Probability of $8 \times$ "$\texttt{right}$": $1/2^8$, time $\times 4$.
  Turn our argument with the rotation!

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

- Estimate the lifetime of a particular game piece.
- ...
- 4. Pretend: only one player, allow rotation.
  Probability of $8 \times$ "`right`": $1/2^8$, time $\times 4$.
  Turn our argument with the rotation!
- 5. Pretend: nothing.
  Probability of $8 \times$ "`right`": $1/2^8$, time $\times 8$.

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

- Estimate the lifetime of a particular game piece.
- ...
- 5. Pretend: nothing.
  Probability of $8 \times$ "`right`": $1/2^8$, time $\times 8$.
- Conclusion: a piece's expected lifetime is $2^8 \cdot 8 = 2048$ moves (upper bound).

# Board Blitz — Solution 2

*In this problem, you have to quickly find the final position in a game on a square board.*

- Estimate the lifetime of a particular game piece.
- ...
- 5. Pretend: nothing.
  Probability of $8 \times$ "`right`": $1/2^8$, time $\times 8$.
- Conclusion: a piece's expected lifetime is $2^8 \cdot 8 = 2048$ moves (upper bound).
- Simulate only the last 10 000 moves by slow but simple code.

# Divisor Count — Preliminaries

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

# Divisor Count — Preliminaries

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

- Consider all factorizations of $k$: $k = a_1 \cdot a_2 \cdot \ldots \cdot a_r$.

# Divisor Count — Preliminaries

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

- Consider all factorizations of $k$: $k = a_1 \cdot a_2 \cdot \ldots \cdot a_r$.
- Number with $k$ divisors: $p_1^{a_1-1} \cdot p_2^{a_2-1} \cdot \ldots \cdot p_r^{a_r-1}$ ($p_i$ are primes).

# Divisor Count — Preliminaries

*In this problem, you have to find the first n integers which have exactly k divisors.*

- Consider all factorizations of $k$: $k = a_1 \cdot a_2 \cdot \ldots \cdot a_r$.
- Number with $k$ divisors: $p_1^{a_1-1} \cdot p_2^{a_2-1} \cdot \ldots \cdot p_r^{a_r-1}$ ($p_i$ are primes).

Minimum such number?

# Divisor Count — Preliminaries

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

- Consider all factorizations of $k$: $k = a_1 \cdot a_2 \cdot \ldots \cdot a_r$.
- Number with $k$ divisors: $p_1^{a_1-1} \cdot p_2^{a_2-1} \cdot \ldots \cdot p_r^{a_r-1}$ ($p_i$ are primes).

Minimum such number?

- Consecutive primes: $p_1 = 2$, $p_2 = 3$, ....

# Divisor Count — Preliminaries

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

- Consider all factorizations of $k$: $k = a_1 \cdot a_2 \cdot \ldots \cdot a_r$.
- Number with $k$ divisors: $p_1^{a_1-1} \cdot p_2^{a_2-1} \cdot \ldots \cdot p_r^{a_r-1}$ ($p_i$ are primes).

Minimum such number?

- Consecutive primes: $p_1 = 2$, $p_2 = 3$, $\ldots$.
- Non-increasing powers: $a_1 \geqslant a_2 \geqslant \ldots \geqslant a_r$.

# Divisor Count — Example 1

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

# Divisor Count — Example 1

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Example: $k = 12$.

- $12 = 12$
  $x = 2^{11} = 2048$

# Divisor Count — Example 1

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Example: $k = 12$.

- $12 = 12$
  $x = 2^{11} = 2048$
- $12 = 6 \cdot 2$
  $x = 2^5 \cdot 3^1 = 96$

# Divisor Count — Example 1

*In this problem, you have to find the first n integers which have exactly k divisors.*

Example: $k = 12$.

- $12 = 12$
  $x = 2^{11} = 2048$
- $12 = 6 \cdot 2$
  $x = 2^5 \cdot 3^1 = 96$
- $12 = 4 \cdot 3$
  $x = 2^3 \cdot 3^2 = 72$

# Divisor Count — Example 1

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Example: $k = 12$.

- $12 = 12$
  $x = 2^{11} = 2048$
- $12 = 6 \cdot 2$
  $x = 2^5 \cdot 3^1 = 96$
- $12 = 4 \cdot 3$
  $x = 2^3 \cdot 3^2 = 72$
- $12 = 3 \cdot 2 \cdot 2$
  $x = 2^2 \cdot 3^1 \cdot 5^1 = 60$

# Divisor Count — Example 1

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Example: $k = 12$.

- $12 = 12$
  $x = 2^{11} = 2048$
- $12 = 6 \cdot 2$
  $x = 2^5 \cdot 3^1 = 96$
- $12 = 4 \cdot 3$
  $x = 2^3 \cdot 3^2 = 72$
- $12 = 3 \cdot 2 \cdot 2$
  $x = 2^2 \cdot 3^1 \cdot 5^1 = 60$

We compare the numbers by putting them into a priority queue.

# Divisor Count — Generating

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Generating next numbers:

- $2^{11} \longrightarrow 3^{11} \longrightarrow 5^{11} \longrightarrow 7^{11} \longrightarrow 11^{11} \longrightarrow 13^{11} \longrightarrow \ldots$

# Divisor Count — Generating

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Generating next numbers:

- $2^{11} \longrightarrow 3^{11} \longrightarrow 5^{11} \longrightarrow 7^{11} \longrightarrow 11^{11} \longrightarrow 13^{11} \longrightarrow \ldots$
- $2^5 \cdot 3^1 \longrightarrow 2^1 \cdot 3^5$ and $2^5 \cdot 5^1$

# Divisor Count — Generating

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Generating next numbers:

- $2^{11} \longrightarrow 3^{11} \longrightarrow 5^{11} \longrightarrow 7^{11} \longrightarrow 11^{11} \longrightarrow 13^{11} \longrightarrow \ldots$
- $2^5 \cdot 3^1 \longrightarrow 2^1 \cdot 3^5$ and $2^5 \cdot 5^1$
- $2^3 \cdot 3^2 \longrightarrow 2^2 \cdot 3^3$ and $2^3 \cdot 5^2$

# Divisor Count — Generating

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Generating next numbers:

- $2^{11} \longrightarrow 3^{11} \longrightarrow 5^{11} \longrightarrow 7^{11} \longrightarrow 11^{11} \longrightarrow 13^{11} \longrightarrow \dots$
- $2^5 \cdot 3^1 \longrightarrow 2^1 \cdot 3^5$ and $2^5 \cdot 5^1$
- $2^3 \cdot 3^2 \longrightarrow 2^2 \cdot 3^3$ and $2^3 \cdot 5^2$
- $2^2 \cdot 3^1 \cdot 5^1 \longrightarrow 2^2 \cdot 3^1 \cdot 7^1$ and $2^1 \cdot 3^2 \cdot 5^1$
    - $2^2 \cdot 3^1 \cdot 7^1 \longrightarrow 2^2 \cdot 5^1 \cdot 7^1$ and $2^2 \cdot 3^1 \cdot 11^1$ and $2^1 \cdot 3^2 \cdot 7^1$
    - $2^1 \cdot 3^2 \cdot 5^1 \longrightarrow 2^1 \cdot 3^2 \cdot 7^1$ and $2^1 \cdot 3^1 \cdot 5^2$
    - $\dots$

# Divisor Count — Example 2

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

A more complex example:

- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ produces:

# Divisor Count — Example 2

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

A more complex example:

- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ produces:
- $3^3 \cdot 7^1 \cdot 11^3 \cdot 17^2 \cdot 29^1$ (swapping adjacent powers 2 and 3)

# Divisor Count — Example 2

*In this problem, you have to find the first n integers which have exactly k divisors.*

A more complex example:

- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ produces:
- $3^3 \cdot 7^1 \cdot 11^3 \cdot 17^2 \cdot 29^1$ (swapping adjacent powers 2 and 3)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^1 \cdot 29^2$ (swapping adjacent powers 4 and 5)

# Divisor Count — Example 2

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

A more complex example:

- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ produces:
- $3^3 \cdot 7^1 \cdot 11^3 \cdot 17^2 \cdot 29^1$ (swapping adjacent powers 2 and 3)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^1 \cdot 29^2$ (swapping adjacent powers 4 and 5)
- $5^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ (shifting element 1 to next prime)

# Divisor Count — Example 2

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

A more complex example:

- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ produces:
- $3^3 \cdot 7^1 \cdot 11^3 \cdot 17^2 \cdot 29^1$ (swapping adjacent powers 2 and 3)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^1 \cdot 29^2$ (swapping adjacent powers 4 and 5)
- $5^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ (shifting element 1 to next prime)
- $3^3 \cdot 7^3 \cdot 13^1 \cdot 17^2 \cdot 29^1$ (shifting element 3 to next prime)

## Divisor Count — Example 2

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

A more complex example:

- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ produces:
- $3^3 \cdot 7^1 \cdot 11^3 \cdot 17^2 \cdot 29^1$ (swapping adjacent powers 2 and 3)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^1 \cdot 29^2$ (swapping adjacent powers 4 and 5)
- $5^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ (shifting element 1 to next prime)
- $3^3 \cdot 7^3 \cdot 13^1 \cdot 17^2 \cdot 29^1$ (shifting element 3 to next prime)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 23^2 \cdot 29^1$ (shifting element 4 to next prime)

# Divisor Count — Example 2

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

A more complex example:

- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ produces:
- $3^3 \cdot 7^1 \cdot 11^3 \cdot 17^2 \cdot 29^1$ (swapping adjacent powers 2 and 3)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^1 \cdot 29^2$ (swapping adjacent powers 4 and 5)
- $5^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ (shifting element 1 to next prime)
- $3^3 \cdot 7^3 \cdot 13^1 \cdot 17^2 \cdot 29^1$ (shifting element 3 to next prime)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 23^2 \cdot 29^1$ (shifting element 4 to next prime)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 31^1$ (shifting element 5 to next prime)

# Divisor Count — Example 2

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

A more complex example:

- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ produces:
- $3^3 \cdot 7^1 \cdot 11^3 \cdot 17^2 \cdot 29^1$ (swapping adjacent powers 2 and 3)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^1 \cdot 29^2$ (swapping adjacent powers 4 and 5)
- $5^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 29^1$ (shifting element 1 to next prime)
- $3^3 \cdot 7^3 \cdot 13^1 \cdot 17^2 \cdot 29^1$ (shifting element 3 to next prime)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 23^2 \cdot 29^1$ (shifting element 4 to next prime)
- $3^3 \cdot 7^3 \cdot 11^1 \cdot 17^2 \cdot 31^1$ (shifting element 5 to next prime)

Priority queue size limit: the remaining number of values to output.

# Divisor Count — Tests

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Interesting test cases:

# Divisor Count — Tests

*In this problem, you have to find the first $n$ integers which have exactly $k$ divisors.*

Interesting test cases:

- $k = 103\,680$:
  $897\,612\,484\,786\,617\,600,\ 994\,651\,672\,331\,116\,800,\ -1,\ -1,\ \ldots.$

# Divisor Count — Tests

*In this problem, you have to find the first n integers which have exactly k divisors.*

Interesting test cases:

- $k = 103\,680$:
  $897\,612\,484\,786\,617\,600,\ 994\,651\,672\,331\,116\,800,\ -1,\ -1,\ \ldots$.
- $k = 11\,520$:
  initial priority queue size is 201.

# Divisor Count — Tests

*In this problem, you have to find the first n integers which have exactly k divisors.*

Interesting test cases:

- $k = 103\,680$:
  $897\,612\,484\,786\,617\,600,\ 994\,651\,672\,331\,116\,800,\ -1,\ -1,\ \ldots$.
- $k = 11\,520$:
  initial priority queue size is 201.
- $k = 1$:
  the only such number is 1.

# Domino Tiling — Outline

*In this problem, you have to transform one domino tiling of a rectangle into another by moving dominoes along cycles.*

# Domino Tiling — Outline

*In this problem, you have to transform one domino tiling of a rectangle into another by moving dominoes along cycles.*

- If a domino is present in both tilings, we will not touch it.

# Domino Tiling — Outline

*In this problem, you have to transform one domino tiling of a rectangle into another by moving dominoes along cycles.*

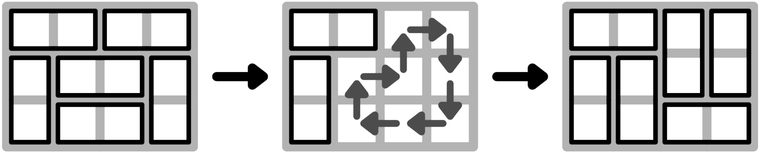
- If a domino is present in both tilings, we will not touch it.
- If a cell belongs to different dominoes, we have to touch it at least once.

# Domino Tiling — Outline

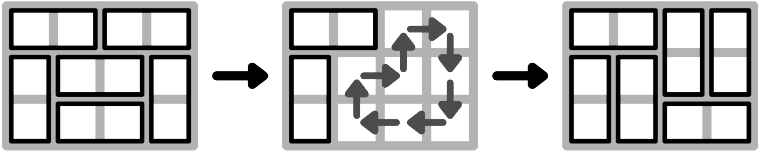
*In this problem, you have to transform one domino tiling of a rectangle into another by moving dominoes along cycles.*

- If a domino is present in both tilings, we will not touch it.
- If a cell belongs to different dominoes, we have to touch it at least once.
- At least once $\longrightarrow$ exactly once.

# Domino Tiling — Example



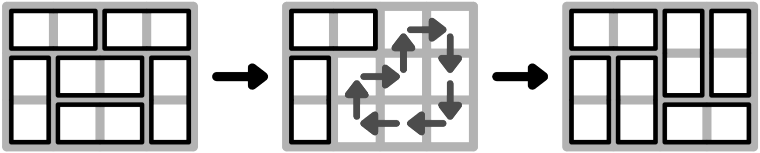
# Domino Tiling — Example



One step of the algorithm:

- Pick a cell which belongs to different dominoes.

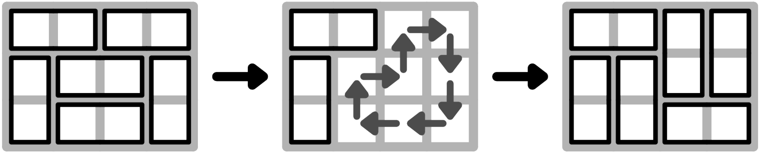
# Domino Tiling — Example



One step of the algorithm:

- Pick a cell which belongs to different dominoes.
- Move to another cell along the domino on board 1.

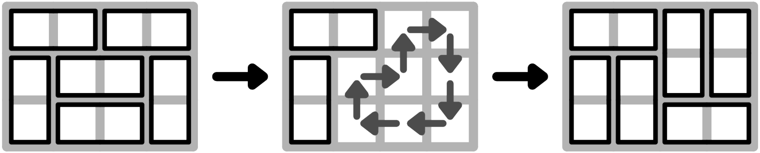
# Domino Tiling — Example



One step of the algorithm:

- Pick a cell which belongs to different dominoes.
- Move to another cell along the domino on board 1.
- Move to yet another cell along the domino on board 2.

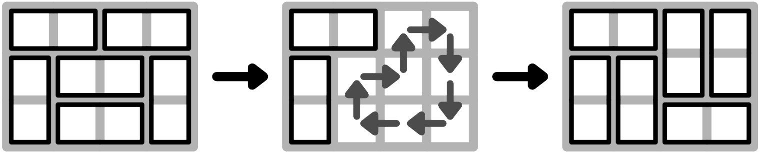
# Domino Tiling — Example



One step of the algorithm:

- Pick a cell which belongs to different dominoes.
- Move to another cell along the domino on board 1.
- Move to yet another cell along the domino on board 2.
- ...

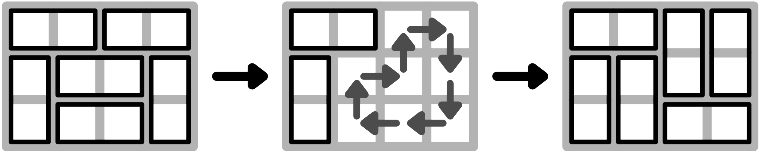
# Domino Tiling — Example



One step of the algorithm:

- Pick a cell which belongs to different dominoes.
- Move to another cell along the domino on board 1.
- Move to yet another cell along the domino on board 2.
- ...
- The process could have started from board 2 (in reverse).

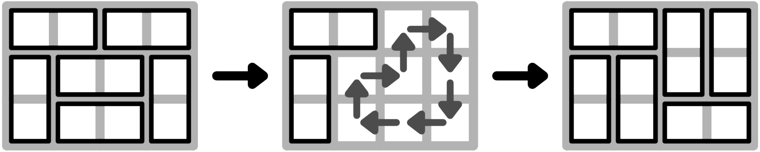
# Domino Tiling — Example



One step of the algorithm:

- Pick a cell which belongs to different dominoes.
- Move to another cell along the domino on board 1.
- Move to yet another cell along the domino on board 2.
- ...
- The process could have started from board 2 (in reverse).
- So, we will get a cycle.

# Domino Tiling — Example



One step of the algorithm:

- Pick a cell which belongs to different dominoes.
- Move to another cell along the domino on board 1.
- Move to yet another cell along the domino on board 2.
- ...
- The process could have started from board 2 (in reverse).
- So, we will get a cycle.
- Move the dominoes along this cycle once to remove it completely.

# Hourglass — Solution?

*In this problem, you have to solve a puzzle involving a planar board and pieces.*

# Hourglass — Solution?

*In this problem, you have to solve a puzzle involving a planar board and pieces.*

How to solve it?

# Hourglass — Solution?

*In this problem, you have to solve a puzzle involving a planar board and pieces.*

How to solve it?

Constructive solution?
What if $n$ is up to 100?

# Hourglass — Brute Force

*In this problem, you have to solve a puzzle involving a planar board and pieces.*

# Hourglass — Brute Force

*In this problem, you have to solve a puzzle involving a planar board and pieces.*

- Key: the order in which we try the moves

# Hourglass — Brute Force

*In this problem, you have to solve a puzzle involving a planar board and pieces.*

- Key: the order in which we try the moves
- Pruning: **polynomial** check vs. **exponential** branching

# Hourglass — Brute Force

*In this problem, you have to solve a puzzle involving a planar board and pieces.*

- Key: the order in which we try the moves
- Pruning: **polynomial** check vs. **exponential** branching
- DFS (brute force) or BFS (beam search)

# Hourglass — Brute Force

*In this problem, you have to solve a puzzle involving a planar board and pieces.*

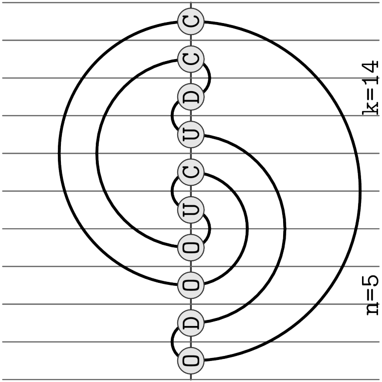
- Key: the order in which we try the moves
- Pruning: **polynomial** check vs. **exponential** branching
- DFS (brute force) or BFS (beam search)
- Simplification: consider 1D version

# $k$-th Meander — Section

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

# *k*-th Meander — Section

*In this problem, you have to find the lexicographically k-th meander of order n.*




- .
- [.]
- [].[]
- [][.]
- [].[]
- [[].][]
- [[.]][]
- [.][]
- .[]
- [.]
- .

# $k$-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

# $k$-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

- State: bracket sequence (with length) and intersection point.

# $k$-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

- State: bracket sequence (with length) and intersection point.
- Transitions: one per `O`, `D`, `U`, `C`.

# $k$-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

- State: bracket sequence (with length) and intersection point.
- Transitions: one per `O`, `D`, `U`, `C`.
- The number of meanders for $n = 15$ is $602\,188\,541\,928$.

# $k$-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

- State: bracket sequence (with length) and intersection point.
- Transitions: one per `O`, `D`, `U`, `C`.
- The number of meanders for $n = 15$ is $602\,188\,541\,928$.
- It is $> 6$ times larger than $10^{11}$.

# *k*-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically k-th meander of order n.*

- State: bracket sequence (with length) and intersection point.
- Transitions: one per `O`, `D`, `U`, `C`.
- The number of meanders for $n = 15$ is 602 188 541 928.
- It is $> 6$ times larger than $10^{11}$.
- Lazy dynamic programming: create only the states we need.

# $k$-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

- State: bracket sequence (with length) and intersection point.
- Transitions: one per `O`, `D`, `U`, `C`.
- The number of meanders for $n = 15$ is 602 188 541 928.
- It is $> 6$ times larger than $10^{11}$.
- Lazy dynamic programming: create only the states we need.
  - First, recursively find the number $c$ of child states if we make the current transition.

# *k*-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically k-th meander of order n.*

- State: bracket sequence (with length) and intersection point.
- Transitions: one per `O`, `D`, `U`, `C`.
- The number of meanders for $n = 15$ is 602 188 541 928.
- It is $> 6$ times larger than $10^{11}$.
- Lazy dynamic programming: create only the states we need.
  - First, recursively find the number $c$ of child states if we make the current transition.
  - If the number $k > c$, decrease it by $c$ and proceed with the next transition ($\mathtt{C} < \mathtt{D} < \mathtt{O} < \mathtt{U}$).

# $k$-th Meander — Dynamic Programming

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

- State: bracket sequence (with length) and intersection point.
- Transitions: one per `O`, `D`, `U`, `C`.
- The number of meanders for $n = 15$ is $602\,188\,541\,928$.
- It is $> 6$ times larger than $10^{11}$.
- Lazy dynamic programming: create only the states we need.
  - First, recursively find the number $c$ of child states if we make the current transition.
  - If the number $k > c$, decrease it by $c$ and proceed with the next transition ($\mathtt{C} < \mathtt{D} < \mathtt{O} < \mathtt{U}$).
  - Otherwise, print the transition sign and descend by the current transition.

# $k$-th Meander — Reference

*In this problem, you have to find the lexicographically $k$-th meander of order $n$.*

An article on the topic:

- Iwan Jensen, Enumerations of plane meanders: `http://arxiv.org/abs/cond-mat/9910313`

# Records and Cycles — Preliminaries

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

# Records and Cycles — Preliminaries

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Records:
  $f(n, k) = f(n-1, k-1) + (n-1) \cdot f(n-1, k)$.

# Records and Cycles — Preliminaries

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Records:
  $f(n, k) = f(n-1, k-1) + (n-1) \cdot f(n-1, k)$.
- Cycles:
  $g(n, k) = g(n-1, k-1) + (n-1) \cdot g(n-1, k)$.

# Records and Cycles — Solution 1

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

# Records and Cycles — Solution 1

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Let us transform one cycle into one record.

# Records and Cycles — Solution 1

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Let us transform one cycle into one record.
- Find the cycle containing 1, let its length be $l$.

# Records and Cycles — Solution 1

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Let us transform one cycle into one record.
- Find the cycle containing 1, let its length be $l$.
- Put $l$ on the left and 1, 2, $\ldots$, $(l-1)$ after it in their order on the cycle.

# Records and Cycles — Solution 1

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Let us transform one cycle into one record.
- Find the cycle containing 1, let its length be $l$.
- Put $l$ on the left and 1, 2, ..., $(l - 1)$ after it in their order on the cycle.
- Next, find the cycle containing the remaining minimum, let its length be $m$.

# Records and Cycles — Solution 1

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Let us transform one cycle into one record.
- Find the cycle containing 1, let its length be $l$.
- Put $l$ on the left and 1, 2, ..., $(l - 1)$ after it in their order on the cycle.
- Next, find the cycle containing the remaining minimum, let its length be $m$.
- Put $l + m$ on the left and $(l + 1)$, $(l + 2)$, ..., $(l + m - 1)$ after it in their order on the cycle.

# Records and Cycles — Solution 1

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Let us transform one cycle into one record.
- Find the cycle containing 1, let its length be $l$.
- Put $l$ on the left and 1, 2, ..., $(l - 1)$ after it in their order on the cycle.
- Next, find the cycle containing the remaining minimum, let its length be $m$.
- Put $l + m$ on the left and $(l + 1)$, $(l + 2)$, ..., $(l + m - 1)$ after it in their order on the cycle.
- ...

# Records and Cycles — Solution 1

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

- Let us transform one cycle into one record.
- Find the cycle containing 1, let its length be $l$.
- Put $l$ on the left and 1, 2, ..., $(l-1)$ after it in their order on the cycle.
- Next, find the cycle containing the remaining minimum, let its length be $m$.
- Put $l+m$ on the left and $(l+1)$, $(l+2)$, ..., $(l+m-1)$ after it in their order on the cycle.
- ...
- Can be done in linear total time.

# Records and Cycles — Cycles to Records

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform cycles into records ($p \longrightarrow u$).

# Records and Cycles — Cycles to Records

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform cycles into records ($p \longrightarrow u$).

- Suppose we already placed $s$ elements.
  Find the *least* unprocessed element $r$.

# Records and Cycles — Cycles to Records

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform cycles into records ($p \longrightarrow u$).

- Suppose we already placed $s$ elements.
  Find the *least* unprocessed element $r$.
- Find its cycle $(r,\ p(r),\ p(p(r)),\ \ldots)$.
  Let its length be $l$.

# Records and Cycles — Cycles to Records

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform cycles into records ($p \longrightarrow u$).

- Suppose we already placed $s$ elements.
  Find the *least* unprocessed element $r$.
- Find its cycle $(r, p(r), p(p(r)), \ldots)$.
  Let its length be $l$.
- Put numbers $s + l$, $s + l - 1$, …, $s + 1$
  into $u$ at places $r$, $p(r)$, $p(p(r))$.

# Records and Cycles — Cycles to Records

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform cycles into records ($p \longrightarrow u$).

- Suppose we already placed $s$ elements.
  Find the *least* unprocessed element $r$.
- Find its cycle $(r, p(r), p(p(r)), \ldots)$.
  Let its length be $l$.
- Put numbers $s + l$, $s + l - 1$, ..., $s + 1$
  into $u$ at places $r$, $p(r)$, $p(p(r))$.
- Repeat until $s = n$.

# Records and Cycles — Records to Cycles

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform records into cycles ($u \longrightarrow p$).

# Records and Cycles — Records to Cycles

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform records into cycles ($u \longrightarrow p$).

- Suppose we already processed $s$ elements.
  Find the *leftmost* unprocessed element $r$.

# Records and Cycles — Records to Cycles

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform records into cycles ($u \longrightarrow p$).

- Suppose we already processed $s$ elements.
  Find the *leftmost* unprocessed element $r$.
- Find the positions of elements $r$, $r-1$, ..., $s+1$.
  Let them be $v(r)$, $v(r-1)$, ..., $v(s+1)$.

# Records and Cycles — Records to Cycles

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform records into cycles ($u \longrightarrow p$).

- Suppose we already processed $s$ elements.
  Find the *leftmost* unprocessed element $r$.
- Find the positions of elements $r$, $r-1$, …, $s+1$.
  Let them be $v(r)$, $v(r-1)$, …, $v(s+1)$.
- Put numbers $v(r-1)$, $v(r-2)$, …, $v(s+1)$, $v(r)$
  into $p$ at places $v(r)$, $v(r-1)$, …, $v(s+2)$, $v(s+1)$.

# Records and Cycles — Records to Cycles

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

Let us transform records into cycles ($u \longrightarrow p$).

- Suppose we already processed $s$ elements.
  Find the *leftmost* unprocessed element $r$.
- Find the positions of elements $r$, $r-1$, ..., $s+1$.
  Let them be $v(r)$, $v(r-1)$, ..., $v(s+1)$.
- Put numbers $v(r-1)$, $v(r-2)$, ..., $v(s+1)$, $v(r)$
  into $p$ at places $v(r)$, $v(r-1)$, ..., $v(s+2)$, $v(s+1)$.
- Repeat until $s = n$.

# Records and Cycles — Records to Cycles

Example $C \to R$:

```
p = 2 6 8 5 4 3 7 1
p = (1 2 6 3 8) (4 5) (7)
q = 8 1 6 5 4 2 7 3
u = . . . . . . . .
u = 5 4 2 . . 3 . 1
u = 5 4 2 7 6 3 . 1
u = 5 4 2 7 6 3 8 1
```

Example $R \to C$:

```
u = 5 4 2 7 6 3 8 1
records: 5 7 8
v = 8 3 6 2 1 5 4 7
p = . . . . . . . . [5 4 3 2 1 on places 1 2 6 3 8]
p = 2 6 8 . . 3 . 1 [7 6 on places 4 5]
p = 2 6 8 5 4 3 . 1 [8 on place 7]
p = 2 6 8 5 4 3 7 1
```

# Records and Cycles — Solution 2

*In this problem, you have to construct a bijection which maps every permutation of length n with k cycles to a permutation of length n with k records.*

The only requirement is to have no condradictions.

# Records and Cycles — Solution 2

*In this problem, you have to construct a bijection which maps every permutation of length n with k cycles to a permutation of length n with k records.*

The only requirement is to have no condradictions.

- Random permutation with $k$ records/cycles?

# Records and Cycles — Solution 2

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

The only requirement is to have no condradictions.

- Random permutation with $k$ records/cycles?
- $f(n, k) = f(n-1, k-1) + (n-1) \cdot f(n-1, k)$

# Records and Cycles — Solution 2

*In this problem, you have to construct a bijection which maps every permutation of length $n$ with $k$ cycles to a permutation of length $n$ with $k$ records.*

The only requirement is to have no condradictions.

- Random permutation with $k$ records/cycles?
- $f(n, k) = f(n-1, k-1) + (n-1) \cdot f(n-1, k)$
- Next permutation with $k$ records/cycles?

# Rock-Paper-Scissors walk — Field

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

# Rock-Paper-Scissors walk — Field

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to generate a cell?

- $s_1 = s_0 \cdot a + c$

# Rock-Paper-Scissors walk — Field

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to generate a cell?

- $s_1 = s_0 \cdot a + c$
- $\begin{pmatrix} s_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$

# Rock-Paper-Scissors walk — Field

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to generate a cell?

- $s_1 = s_0 \cdot a + c$
- $\begin{pmatrix} s_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- $\begin{pmatrix} s_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$

# Rock-Paper-Scissors walk — Field

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to generate a cell?

- $s_1 = s_0 \cdot a + c$
- $\begin{pmatrix} s_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- $\begin{pmatrix} s_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- 1. Naive: $O(h \cdot w)$.

# Rock-Paper-Scissors walk — Field

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to generate a cell?

- $s_1 = s_0 \cdot a + c$
- $\begin{pmatrix} s_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- $\begin{pmatrix} s_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- 1. Naive: $O(h \cdot w)$.
- 2. Repeated squaring: $O(\log(h \cdot w))$.

# Rock-Paper-Scissors walk — Field

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to generate a cell?

- $s_1 = s_0 \cdot a + c$
- $\begin{pmatrix} s_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- $\begin{pmatrix} s_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- 1. Naive: $O(h \cdot w)$.
- 2. Repeated squaring: $O(\log(h \cdot w))$.
- 3. Blunt: $O(1)$ with $O(h + w)$ precalculation.

# Rock-Paper-Scissors walk — Field

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to generate a cell?

- $s_1 = s_0 \cdot a + c$
- $\begin{pmatrix} s_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- $\begin{pmatrix} s_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- 1. Naive: $O(h \cdot w)$.
- 2. Repeated squaring: $O(\log(h \cdot w))$.
- 3. Blunt: $O(1)$ with $O(h + w)$ precalculation.
- Another method: linear functions instead of matrices.

# Rock-Paper-Scissors walk — Path

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

# Rock-Paper-Scissors walk — Path

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to find a path?

- Just find the closest cell each time.

# Rock-Paper-Scissors walk — Path

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to find a path?

- Just find the closest cell each time.
- Don't forget to look back!

# Rock-Paper-Scissors walk — Path

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to find a path?

- Just find the closest cell each time.
- Don't forget to look back!
- The board can be $100 \times 65\,536$ or $65\,536 \times 100$.

# Rock-Paper-Scissors walk — Path

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to find a path?

- Just find the closest cell each time.
- Don't forget to look back!
- The board can be $100 \times 65\,536$ or $65\,536 \times 100$.
- $\left(\frac{2}{3}\right)^{100} = 2.459654... \cdot 10^{-18}$ but $\left(\frac{2}{3}\right)^{10} = 0.017341...$.

## Rock-Paper-Scissors walk — Path

*In this problem, you have to travel from one corner of a checkered field to the opposite corner conforming to certain rules and avoiding making too much moves.*

How to find a path?

- Just find the closest cell each time.
- Don't forget to look back!
- The board can be $100 \times 65\,536$ or $65\,536 \times 100$.
- $\left(\frac{2}{3}\right)^{100} = 2.459654... \cdot 10^{-18}$ but $\left(\frac{2}{3}\right)^{10} = 0.017341...$.
- Speedup: visit each row from right to left, `break` if found.

# Set Operations — Grouping

*In this problem, you have to perform operations with sets according to the given expression.*

## Set Operations — Grouping

*In this problem, you have to perform operations with sets according to the given expression.*

Consider eight ($2^3$) possible classes of elements.

# Set Operations — Grouping

*In this problem, you have to perform operations with sets according to the given expression.*

Consider eight ($2^3$) possible classes of elements.

| **element** | **in** $A$ | **in** $B$ | **in** $C$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |

# Set Operations — Grouping

*In this problem, you have to perform operations with sets according to the given expression.*

Consider eight ($2^3$) possible classes of elements.

- Solve the problem once for $A = \{4, 5, 6, 7\}$, $B = \{2, 3, 6, 7\}$ and $C = \{1, 3, 5, 7\}$.

# Set Operations — Grouping

*In this problem, you have to perform operations with sets according to the given expression.*

Consider eight ($2^3$) possible classes of elements.

- Solve the problem once for $A = \{4, 5, 6, 7\}$, $B = \{2, 3, 6, 7\}$ and $C = \{1, 3, 5, 7\}$.
- For each given triple $(A, B, C)$, distribute the elements $\{1, 2, \dots, n\}$ into eight classes.

# Set Operations — Grouping

*In this problem, you have to perform operations with sets according to the given expression.*

Consider eight ($2^3$) possible classes of elements.

- Solve the problem once for $A = \{4, 5, 6, 7\}$, $B = \{2, 3, 6, 7\}$ and $C = \{1, 3, 5, 7\}$.
- For each given triple $(A, B, C)$, distribute the elements $\{1, 2, \ldots, n\}$ into eight classes.
- For each class, we know whether it is present in the answer.

# Set Operations — Grouping

*In this problem, you have to perform operations with sets according to the given expression.*

Consider eight ($2^3$) possible classes of elements.

- Solve the problem once for $A = \{4, 5, 6, 7\}$, $B = \{2, 3, 6, 7\}$ and $C = \{1, 3, 5, 7\}$.
- For each given triple $(A, B, C)$, distribute the elements $\{1, 2, \dots, n\}$ into eight classes.
- For each class, we know whether it is present in the answer.
- Result: $O(|S| + t)$ instead of $O(|S| \cdot t)$.

Contest Developers:

- Natalya Ginzburg
- Ivan Kazmenko
- Pavel Kunyavskiy

Questions?