Bitwise
Bitwise

Given:

- Sequence of $N$ integers: $A_1, A_2, \ldots, A_n$
- The integers is forming a circle
- The sequence is divided (partitioned) into $K$ sections
- $\text{power}(\text{section}) = \text{the bitwise OR of all integers in that section}$

Determine:

- The maximum bitwise AND of the powers of the sections in an optimal partition of the circle of integers

$1 \leq K \leq N \leq 5 \times 10^5$, $0 \leq A_i \leq 10^9$
Bitwise

Reverse the thinking:

- Given an integer $X$, can you divide the sequence so that the bitwise AND of the powers of the sections is at least $X$?
- Imagine there is a function `can(X)` that can answer the previous question
- Then we can “greedy the answer”:

```c
int ans = 0;
for (int i = 30; i >= 0; i--) {
    int bit = 1 << i;
    if (can(ans | bit)) {
        ans |= bit;
    }
}
printf("%d\n", ans);
```
Bitwise

can(X): How to divide the sequence so that the bitwise AND of the powers of the sections is at least X?

● Simulation:
  ○ Pick a starting point in the sequence and start performing bitwise OR onwards until the accumulator exceeds X, then you found a section.
  ○ From the last point, continue the process to find the next sections until you go back to the starting point.
  ○ See if you managed to find at least K sections?

● How many starting points are there?
  ○ There are at most log(10^9) = 31 different starting points

Total complexity O(N * 31 * 31) = O(N)
Conveyor Belts
Conveyor Belts

Given:

- \( N \) junctions connected by \( M \) conveyor belts
- \( K \) producers located at the first \( K \) junctions
- Producer \( j \) produces a product each minute \( (x \cdot K + j) \) where \( x \geq 0 \) and \( j = 1, 2, \ldots, K \).
- There is a deterministic route from a producer to the warehouse (junction \( N \))
- Each conveyor belt only transports at most one product at any time
- No limit on the number of products at the junctions

Determine:

- Find the maximum number of producers which can be left running such that all the produced products can be delivered to the warehouse

\( 1 \leq K \leq N \leq 300, \ 0 \leq M \leq 1000 \)
Conveyor Belts

Observation:

- This is a graph problem (junction -> node, conveyor belt -> edge)
- How do we encode this constraint in our graph:
  - Each conveyor belt only transports at most one product at any time
- We can encode the “time” dimension by blowing up a junction into $K$ nodes
  - Junction $A$ is represented as $K$ nodes in the graph (node $A$ at time 0, 1, … $K-1$)
    - The time wraps around. That is, time $K$ is equivalent to time 0
  - A conveyor belt connecting from junction $A$ to junction $B$ is represented as
    - $K$ edges: one edge from node $A$ at time $i$ to node $B$ at time $(i + 1) \% K$
Conveyor Belts

Maximum flow solution:

- Add two new nodes (a **source** node and a **sink** node)
- Connect the source node to all K producers
  - Add an edge from the **source** to **Producer i at time i** with **capacity 1**
- Connect the warehouse at all time periods to a sink with **infinite capacity**
  - Add an edge from **Junction N at time i** (for all i = 0..K-1) to the **sink**
- Run **maximum flow** from the source to the sink
  - The maxflow value is the number of producers that can be left running
  - Use **Dinic’s algorithm** to avoid getting time limit exceeded
    - The runtime is proportional to the maxflow value (max = \( K \))