# Untan Programming Contest II

## Problem Analysis

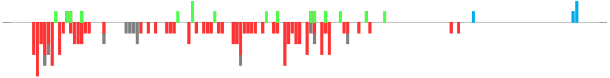
Universitas Tanjungpura

June 2022

# Contest Overview

- Participants: 41 (Out of competition: 3)
- Total Submissions: 292 (AC: 37, Pending: 34)
- Relatively easier than last contest
- No scoreboard resolver, considering using DOMJudge/Kattis next contest

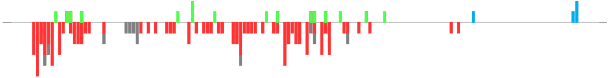
# A: Operasi Aritmatika



**Problem**

Check whether $A \_ B = C$ or $B \_ A = C$ can be true if we can only use $+\ -\ \times\ /$

# A: Operasi Aritmatika



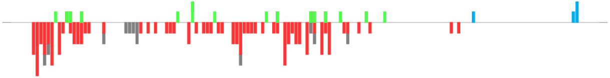
## Problem

Check whether $A \_ B = C$ or $B \_ A = C$ can be true if we can only use $+ \quad - \quad \times \quad /$

## Solution

- Since there are only four operators and two equations, there are at most 6 (six) possible distinct numbers (addition and multiplication is commutative)

# A: Operasi Aritmatika



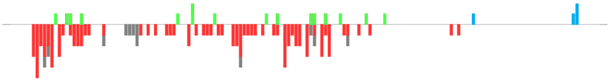
## Problem

Check whether $A \_ B = C$ or $B \_ A = C$ can be true if we can only use $+ \ - \ \times \ /$

## Solution

- Since there are only four operators and two equations, there are at most 6 (six) possible distinct numbers (addition and multiplication is commutative)
- We can try to compute and store all possible numbers in an array or set $S$, then check if $C \in S$

# A: Operasi Aritmatika



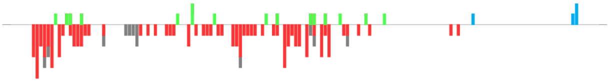
## Problem

Check whether $A \ \_\ B = C$ or $B \ \_\ A = C$ can be true if we can only use $+ \ - \ \times \ /$

## Solution

- Since there are only four operators and two equations, there are at most 6 (six) possible distinct numbers (addition and multiplication is commutative)
- We can try to compute and store all possible numbers in an array or set $S$, then check if $C \in S$
- Be careful of overflow and division by zero when computing these numbers

# A: Operasi Aritmatika



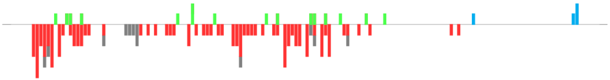
## Problem

Check whether $A \_ B = C$ or $B \_ A = C$ can be true if we can only use $+ \; - \; \times \; /$

## Solution

- Since there are only four operators and two equations, there are at most 6 (six) possible distinct numbers (addition and multiplication is commutative)
- We can try to compute and store all possible numbers in an array or set $S$, then check if $C \in S$
- Be careful of overflow and division by zero when computing these numbers
- Since $C$ is an integer, we should only store the result of $A/B$ if $B|A$ and $B/A$ if $A|B$. Complexity: $O(1)$

# A: Operasi Aritmatika



## Problem

Check whether $A \_ B = C$ or $B \_ A = C$ can be true if we can only use $+ \ - \ \times \ /$

## Solution

- Since there are only four operators and two equations, there are at most 6 (six) possible distinct numbers (addition and multiplication is commutative)
- We can try to compute and store all possible numbers in an array or set $S$, then check if $C \in S$
- Be careful of overflow and division by zero when computing these numbers
- Since $C$ is an integer, we should only store the result of $A/B$ if $B|A$ and $B/A$ if $A|B$. Complexity: $O(1)$

Statistics: 130 submissions, 16 accepted, 4 unknown

# B: Realisasi Wishlist

**Problem**

During promo, what is the minimum amount of money needed to buy all $n$ items?

# B: Realisasi Wishlist

## Problem

During promo, what is the minimum amount of money needed to buy all $n$ items?

## Solution

- Since we're only paying the larger price for each two items, we should really maximize this amount of "not paying"

# B: Realisasi Wishlist

**Problem**

During promo, what is the minimum amount of money needed to buy all $n$ items?

**Solution**

- Since we're only paying the larger price for each two items, we should really maximize this amount of "not paying"
- If you're paying the larger price for each two items, **the second item that's free better be as expensive as possible, possibly not exceeding the first item**

# B: Realisasi Wishlist

## Problem

During promo, what is the minimum amount of money needed to buy all $n$ items?

## Solution

- Since we're only paying the larger price for each two items, we should really maximize this amount of "not paying"
- If you're paying the larger price for each two items, **the second item that's free better be as expensive as possible, possibly not exceeding the first item**
- Therefore, it is always optimal to pick the current most expensive two items and buy them. Let's sort the prices, then sum from the last element to the first with step of 2. Complexity: $O(n \log n)$

# B: Realisasi Wishlist

## Problem

During promo, what is the minimum amount of money needed to buy all $n$ items?

## Solution

- Since we're only paying the larger price for each two items, we should really maximize this amount of "not paying"
- If you're paying the larger price for each two items, **the second item that's free better be as expensive as possible, possibly not exceeding the first item**
- Therefore, it is always optimal to pick the current most expensive two items and buy them. Let's sort the prices, then sum from the last element to the first with step of 2. Complexity: $O(n \log n)$

Statistics: 59 submissions, 8 accepted, 4 unknown

# C: Bilangan Terdekat

**Problem**

What is the smallest integer in the sequence of $1, 2, 3, 4, 8, 9, 16, 27, 32, 64, \ldots$ that is larger than or equal to $x$?

# C: Bilangan Terdekat

## Problem

What is the smallest integer in the sequence of $1, 2, 3, 4, 8, 9, 16, 27, 32, 64, \ldots$ that is larger than or equal to $x$?

## Solution

- If $x \leq 0$, then the answer is $1$. Otherwise, denote $2^n \geq x$ and $3^m \geq x$

# C: Bilangan Terdekat

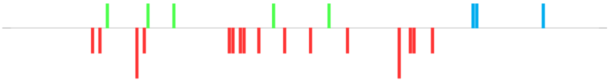
## Problem

What is the smallest integer in the sequence of $1, 2, 3, 4, 8, 9, 16, 27, 32, 64, \ldots$ that is larger than or equal to $x$?

## Solution

- If $x \leq 0$, then the answer is $1$. Otherwise, denote $2^n \geq x$ and $3^m \geq x$
- We seek to minimize $n$ and $m$ and find $\min(2^n, 3^m)$

# C: Bilangan Terdekat



## Problem

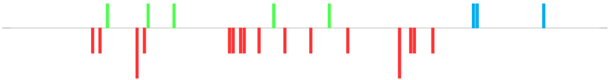
What is the smallest integer in the sequence of $1, 2, 3, 4, 8, 9, 16, 27, 32, 64, \ldots$ that is larger than or equal to $x$?

## Solution

- If $x \leq 0$, then the answer is $1$. Otherwise, denote $2^n \geq x$ and $3^m \geq x$
- We seek to minimize $n$ and $m$ and find $\min(2^n, 3^m)$

$$\begin{aligned} 2^n &\geq x \\ n \log 2 &\geq \log x \\ n &= \left\lceil \frac{\log x}{\log 2} \right\rceil \end{aligned} \qquad \begin{aligned} 3^m &\geq x \\ m \log 3 &\geq \log x \\ m &= \left\lceil \frac{\log x}{\log 3} \right\rceil \end{aligned}$$

# C: Bilangan Terdekat



## Problem

What is the smallest integer in the sequence of $1, 2, 3, 4, 8, 9, 16, 27, 32, 64, \ldots$ that is larger than or equal to $x$?

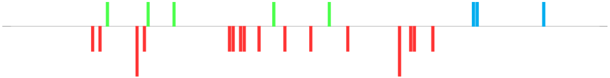
## Solution

- If $x \leq 0$, then the answer is $1$. Otherwise, denote $2^n \geq x$ and $3^m \geq x$
- We seek to minimize $n$ and $m$ and find $\min(2^n, 3^m)$

$$\begin{aligned} 2^n &\geq x \\ n \log 2 &\geq \log x \\ n &= \lceil \frac{\log x}{\log 2} \rceil \end{aligned} \qquad \begin{aligned} 3^m &\geq x \\ m \log 3 &\geq \log x \\ m &= \lceil \frac{\log x}{\log 3} \rceil \end{aligned}$$

- Be careful of precision error when doing this. Complexity: $O(T)$

# C: Bilangan Terdekat



## Problem

What is the smallest integer in the sequence of $1, 2, 3, 4, 8, 9, 16, 27, 32, 64, \ldots$ that is larger than or equal to $x$?

## Alternative Solution

- Store all powers of 2 and 3 in an array $A$. We will need $O(\log_2(10^9) + \log_3(10^9))$ space for this

# C: Bilangan Terdekat

## Problem

What is the smallest integer in the sequence of $1, 2, 3, 4, 8, 9, 16, 27, 32, 64, \ldots$ that is larger than or equal to $x$?

## Alternative Solution

- Store all powers of 2 and 3 in an array $A$. We will need $O(\log_2(10^9) + \log_3(10^9))$ space for this
- For each test we search $A$ to find smallest $y$ such that $y \geq x$. Since $A$ is small, even brute force will work. Complexity: $O(T)$

# C: Bilangan Terdekat

## Problem

What is the smallest integer in the sequence of $1, 2, 3, 4, 8, 9, 16, 27, 32, 64, \ldots$ that is larger than or equal to $x$?

## Alternative Solution

- Store all powers of 2 and 3 in an array $A$. We will need $O(\log_2(10^9) + \log_3(10^9))$ space for this
- For each test we search $A$ to find smallest $y$ such that $y \geq x$. Since $A$ is small, even brute force will work. Complexity: $O(T)$

Statistics: 26 submissions, 5 accepted, 3 unknown

# D: Muzik

**Problem**

Sort the music by their frequency. If they have the same frequency, prioritize the earliest input.

# D: Muzik

## Problem

Sort the music by their frequency. If they have the same frequency, prioritize the earliest input.

## Solution

- First way is to use `stable_sort` since we want to preserve ordering.

# D: Muzik

## Problem

Sort the music by their frequency. If they have the same frequency, prioritize the earliest input.

## Solution

- First way is to use `stable_sort` since we want to preserve ordering.
- The second way is to realize that we can "attach" the original numbering on the input before sorting. So we will store it as (name, frequency, input_num). We first sort by frequency, then by input_num if the frequency is the same. Complexity: $O(n \log n)$

# D: Muzik

## Problem

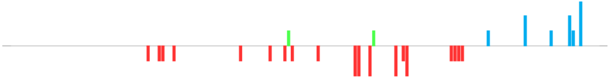
Sort the music by their frequency. If they have the same frequency, prioritize the earliest input.

## Solution

- First way is to use `stable_sort` since we want to preserve ordering.
- The second way is to realize that we can "attach" the original numbering on the input before sorting. So we will store it as (name, frequency, input_num). We first sort by frequency, then by input_num if the frequency is the same. Complexity: $O(n \log n)$

Statistics: 21 submissions, 6 accepted, 6 unknown

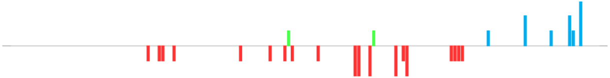
# E: Perbedaan Kelipatan



**Problem**

How many minimum operations needed to make $x$ multiple of $y$ or vice versa?

# E: Perbedaan Kelipatan



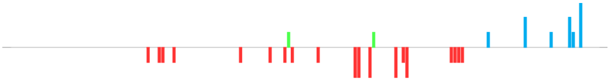
## Problem

How many minimum operations needed to make $x$ multiple of $y$ or vice versa?

## Solution

- Notice that the answer can't exceed $|x - y|$ (make $x$ and $y$ the same). So, we are somewhat assured that brute force is possible.

# E: Perbedaan Kelipatan



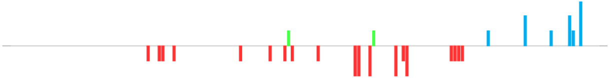
## Problem

How many minimum operations needed to make $x$ multiple of $y$ or vice versa?

## Solution

- Notice that the answer can't exceed $|x - y|$ (make $x$ and $y$ the same). So, we are somewhat assured that brute force is possible.
- Let's say we did exactly $a$ operations on $x$, so $x$ is now $x + a$. Depending on the cases, we may add 0 or more operations on $y$.

# E: Perbedaan Kelipatan



## Problem

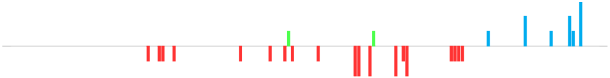
How many minimum operations needed to make $x$ multiple of $y$ or vice versa?

## Solution

- Notice that the answer can't exceed $|x - y|$ (make $x$ and $y$ the same). So, we are somewhat assured that brute force is possible.
- Let's say we did exactly $a$ operations on $x$, so $x$ is now $x + a$. Depending on the cases, we may add 0 or more operations on $y$.
- There are 3 cases:

  $x + a < y$: Do $b$ operations on $y$ (to $y + b$) so $x + a$ divides $y + b$
  $x + a = y$: We don't need any more operations on $y$ since they are the same number
  $x + a > y$: Do $b$ operations on $y$ (to $y + b$) so $y + b$ divides $x + a$

# E: Perbedaan Kelipatan



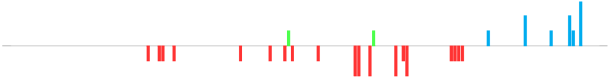
## Problem

How many minimum operations needed to make $x$ multiple of $y$ or vice versa?

## Solution

- Notice that the answer can't exceed $|x - y|$ (make $x$ and $y$ the same). So, we are somewhat assured that brute force is possible.
- Let's say we did exactly $a$ operations on $x$, so $x$ is now $x + a$. Depending on the cases, we may add 0 or more operations on $y$.
- There are 3 cases:
  - $x + a < y$: Do $b$ operations on $y$ (to $y + b$) so $x + a$ divides $y + b$
  - $x + a = y$: We don't need any more operations on $y$ since they are the same number
  - $x + a > y$: Do $b$ operations on $y$ (to $y + b$) so $y + b$ divides $x + a$
- The answer is the minimum of $a + b$. Complexity: $O(|x - y|)$

# E: Perbedaan Kelipatan



## Problem

How many minimum operations needed to make $x$ multiple of $y$ or vice versa?

## Solution

- Notice that the answer can't exceed $|x - y|$ (make $x$ and $y$ the same). So, we are somewhat assured that brute force is possible.
- Let's say we did exactly $a$ operations on $x$, so $x$ is now $x + a$. Depending on the cases, we may add 0 or more operations on $y$.
- There are 3 cases:

  $x + a < y$: Do $b$ operations on $y$ (to $y + b$) so $x + a$ divides $y + b$
  $x + a = y$: We don't need any more operations on $y$ since they are the same number
  $x + a > y$: Do $b$ operations on $y$ (to $y + b$) so $y + b$ divides $x + a$
- The answer is the minimum of $a + b$. Complexity: $O(|x - y|)$

Statistics: 36 submissions, 2 accepted, 10 unknown

# F: Two Water Jug Puzzle

**Problem**

Determine the minimum amount of operations to make both water jugs have the same volume.

# F: Two Water Jug Puzzle

## Problem

Determine the minimum amount of operations to make both water jugs have the same volume.

## Solution

- First, if $a = b$ then the answer is $0$. If only one of them is 0, then it's impossible. Otherwise, it's always possible

# F: Two Water Jug Puzzle

## Problem

Determine the minimum amount of operations to make both water jugs have the same volume.

## Solution

- First, if $a = b$ then the answer is $0$. If only one of them is 0, then it's impossible. Otherwise, it's always possible
- Observation: Drinking $\frac{p-1}{p}$ means there's $\frac{1}{p}$ of it left. If there is $x$ ml initially, then there's $\frac{x}{p}$ left

# F: Two Water Jug Puzzle

### Problem

Determine the minimum amount of operations to make both water jugs have the same volume.

### Solution

- First, if $a = b$ then the answer is $0$. If only one of them is 0, then it's impossible. Otherwise, it's always possible
- Observation: Drinking $\frac{p-1}{p}$ means there's $\frac{1}{p}$ of it left. If there is $x$ ml initially, then there's $\frac{x}{p}$ left
- Now, decompose $a$ and $b$ into their prime factors. To make $a$ and $b$ the same volume, for every prime factor, we have to make the exponent the same. Complexity: $O(\sqrt{a} + \sqrt{b})$

# F: Two Water Jug Puzzle

## Problem

Determine the minimum amount of operations to make both water jugs have the same volume.

## Solution

- First, if $a = b$ then the answer is $0$. If only one of them is 0, then it's impossible. Otherwise, it's always possible
- Observation: Drinking $\frac{p-1}{p}$ means there's $\frac{1}{p}$ of it left. If there is $x$ ml initially, then there's $\frac{x}{p}$ left
- Now, decompose $a$ and $b$ into their prime factors. To make $a$ and $b$ the same volume, for every prime factor, we have to make the exponent the same. Complexity: $O(\sqrt{a} + \sqrt{b})$

Statistics: 4 submissions, 0 accepted, 2 unknown

# G: 7-Jackpot

**Problem**

Choose 7 different boxes so it sums up to exactly 777.

# G: 7-Jackpot

## Problem

Choose 7 different boxes so it sums up to exactly 777.

## Solution

- Since every $a_i$ is distinct, denote $ok[x]$ if $x$ exists

# G: 7-Jackpot

## Problem

Choose 7 different boxes so it sums up to exactly 777.

## Solution

- Since every $a_i$ is distinct, denote $ok[x]$ if $x$ exists
- Iterate from the first item to the sixth item. **The seventh item** must have the value of $a_7 = 777 - a_1 - a_2 - a_3 - a_4 - a_5 - a_6$. This obviously would be TLE.

# G: 7-Jackpot

## Problem

Choose 7 different boxes so it sums up to exactly 777.

## Solution

- Since every $a_i$ is distinct, denote $ok[x]$ if $x$ exists
- Iterate from the first item to the sixth item. **The seventh item** must have the value of $a_7 = 777 - a_1 - a_2 - a_3 - a_4 - a_5 - a_6$. This obviously would be TLE.
- We should stop traversing once we know it's invalid. It's not valid when the sum is larger than 777, or the seventh item is not valid/does not exist.

# G: 7-Jackpot

## Problem

Choose 7 different boxes so it sums up to exactly 777.

## Solution

- Since every $a_i$ is distinct, denote $ok[x]$ if $x$ exists
- Iterate from the first item to the sixth item. **The seventh item** must have the value of $a_7 = 777 - a_1 - a_2 - a_3 - a_4 - a_5 - a_6$. This obviously would be TLE.
- We should stop traversing once we know it's invalid. It's not valid when the sum is larger than 777, or the seventh item is not valid/does not exist.
- Make sure $a_7$ is not the same as the other six box chosen. Complexity: $O(1)$

# G: 7-Jackpot

## Problem

Choose 7 different boxes so it sums up to exactly 777.

## Solution

- Since every $a_i$ is distinct, denote $ok[x]$ if $x$ exists
- Iterate from the first item to the sixth item. **The seventh item** must have the value of $a_7 = 777 - a_1 - a_2 - a_3 - a_4 - a_5 - a_6$. This obviously would be TLE.
- We should stop traversing once we know it's invalid. It's not valid when the sum is larger than 777, or the seventh item is not valid/does not exist.
- Make sure $a_7$ is not the same as the other six box chosen. Complexity: $O(1)$

Statistics: 13 submissions, 0 accepted, 8 unknown

# H: Selamat Berbelanja

**Problem**

Buy one or more items such that you only have to pay using cash (no coin change)

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- First, if there exist an item with an ending $000$, then it's possible

# H: Selamat Berbelanja

**Problem**

Buy one or more items such that you only have to pay using cash (no coin change)

**Solution**

- First, if there exist an item with an ending $000$, then it's possible
- If not, then if $k > 198801$, then it's possible. The proof is left to the reader

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- First, if there exist an item with an ending $000$, then it's possible
- If not, then if $k > 198801$, then it's possible. The proof is left to the reader
- If $k \leq 199801$, we will solve this using dynamic programming

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- Observation: since we only want the total price to have an ending of $000$, only the last three digits matter. So, we will mod all price by $1000$

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- Observation: since we only want the total price to have an ending of $000$, only the last three digits matter. So, we will mod all price by $1000$
- Let $dp[k][i]$ denote that if it's possible to get change $i$ from the first $k$ items

# H: Selamat Berbelanja

**Problem**

Buy one or more items such that you only have to pay using cash (no coin change)

**Solution**

- Observation: since we only want the total price to have an ending of $000$, only the last three digits matter. So, we will mod all price by $1000$
- Let $dp[k][i]$ denote that if it's possible to get change $i$ from the first $k$ items
- The next state $k+1$ will be computed as follows: for all possible changes in previous state, we include it in this state (do not buy the $k+1$-th item)

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- Observation: since we only want the total price to have an ending of $000$, only the last three digits matter. So, we will mod all price by $1000$
- Let $dp[k][i]$ denote that if it's possible to get change $i$ from the first $k$ items
- The next state $k+1$ will be computed as follows: for all possible changes in previous state, we include it in this state (do not buy the $k+1$-th item)
- For each possible state $i$ in $k$, then $i + a_{k+1} \mod 1000$ is also possible

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- Finally, we consider the possibility where we didn't buy the previous $k$ item and only buy this item, so $dp[k+1][a_{k+1} \mod 1000]$ is possible

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- Finally, we consider the possibility where we didn't buy the previous $k$ item and only buy this item, so $dp[k+1][a_{k+1} \mod 1000]$ is possible
- We notice that we only need the previous state, so really the dp dimension is $dp[2][1000]$

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- Finally, we consider the possibility where we didn't buy the previous $k$ item and only buy this item, so $dp[k+1][a_{k+1} \mod 1000]$ is possible
- We notice that we only need the previous state, so really the dp dimension is $dp[2][1000]$
- A solution exists if $dp[j][0]$ is true for any $j$. Complexity: $O(k)$

# H: Selamat Berbelanja

## Problem

Buy one or more items such that you only have to pay using cash (no coin change)

## Solution

- Finally, we consider the possibility where we didn't buy the previous $k$ item and only buy this item, so $dp[k+1][a_{k+1} \mod 1000]$ is possible
- We notice that we only need the previous state, so really the dp dimension is $dp[2][1000]$
- A solution exists if $dp[j][0]$ is true for any $j$. Complexity: $O(k)$

Statistics: 3 submissions, 0 accepted, 0 unknown