TWGSS
OI Team

# 2024/25 HKOI Final Mock I

## Editorial

26 November 2024

# Statistics

| | Problem | Attempts | Full | Max | Mean | Std Dev | LQ | Median | UQ | Subtasks |
|---|---|---|---|---|---|---|---|---|---|---|
| TMF251 | Deer Crackers | 16 | 100 | 100 | 34.563 | 25.483 | 5 | 39 | 51 | 5 \| 16, 12 \| 7, 34 \| 10, 21 \| 1, 28 \| 1 |
| TMF252 | Deer Shop | 16 | 100 | 100 | 51.813 | 39.778 | 9 | 40 | 100 | 9 \| 15, 12 \| 8, 31 \| 10, 26 \| 6, 22 \| 6 |
| TMF253 | Deer Friend | 10 | 100 | 100 | 30.3 | 35.372 | 8 | 19 | 19 | 8 \| 8, 11 \| 7, 18 \| 2, 27 \| 2, 15 \| 2, 21 \| 2 |
| TMF254 | Deer Jane | 9 | 100 | 100 | 45.444 | 40.882 | 7 | 41 | 100 | 7 \| 7, 13 \| 6, 21 \| 5, 19 \| 3, 24 \| 3, 16 \| 3 |
| | | **17** | **400** | **400** | **123.176** | **118.304** | **38.5** | **66** | **211** | |

# Prizes Estimation

This time, the judgement will base on HKOI Junior (less harsh!)

| | | | Gold | Silver | Bronze | HM |
|---|---|---|---|---|---|---|
| Expected Score | *TMF251* | Deer Crackers | 100 | 72 | 51 | 17 |
| | *TMF252* | Deer Shop | 100 | 100 | 78 | 52 |
| | *TMF253* | Deer Friend | 100 | 64 | 37 | 19 |
| | *TMF254* | Deer Jane | 100 | 60 | 41 | 20 |
| | Total | | 400 | 296 | 207 | 108 |
| Multiplier | | | ×0.925 | | | |
| **Cut-off** | | | **370** | **274** | **191** | **100** |
| No. of Recipients | | | 1 | 1 | 3 | 1 |

# Fun Fact

Author gets too high after watching しかのこのこのここしたんたん...

Therefore, every problem in this contest is about Nokotan (のこたん)!

# Deer Crackers

**TMF251**

# Problem Statement

- There are some deer and some deer crackers locate on a number line.
- Assign each deer to a **different** deer cracker.
- **Minimise** the maximum distance between the deer and the assigned cracker.

# Subtask 1

$N = M = 1$

- Output the absolute difference between $D_1$ and $C_1$
- Everyone should get this subtask!
- … Everyone get this subtask, right?

| Expected Score | 5 |
| --- | --- |

| Cumulative | 5 |
| --- | --- |

# Subtask 2

$N = 2$
$M = 3$

- There are just $3^2 = 9$ possible assignments (which only $6$ of them are valid, as the cracker assigned to each deer must be different)!
- Just consider all the possible assignments of deer to crackers and take the minimum of all the answers.
- Simple hard code implementation exercise!

| | |
|---|---|
| Expected Score | **12** |

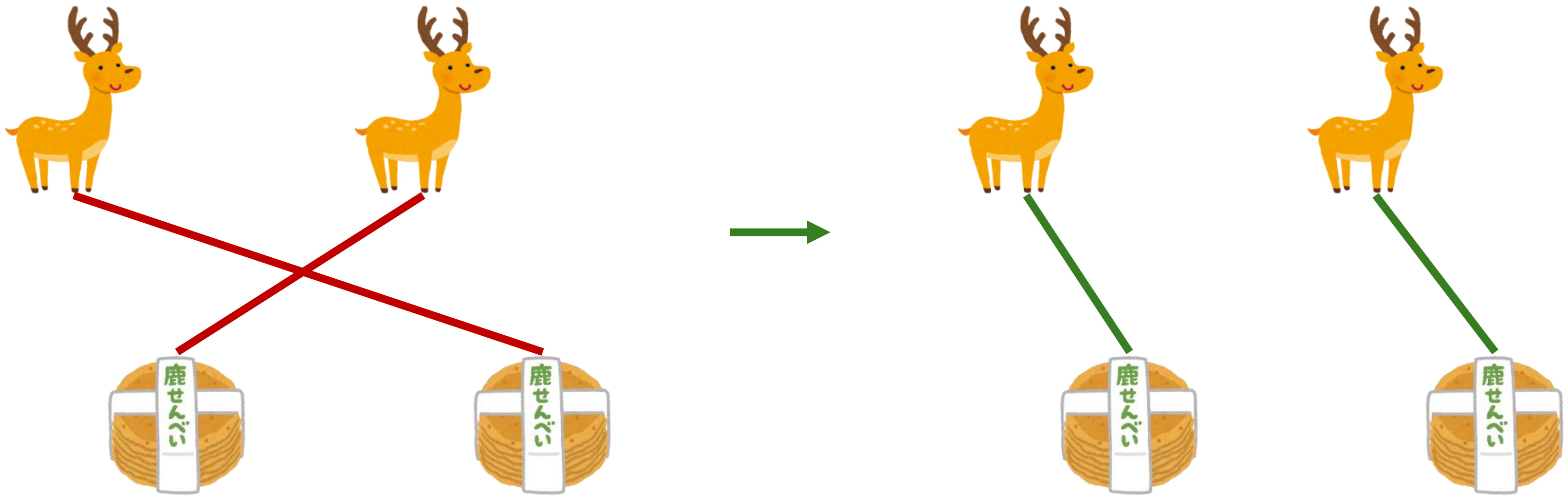
| | |
|---|---|
| Cumulative | 17 |

# Subtask 3

$N = M$

<u>Key Observation</u>

It is always not worse if the route of the deer does not “cross” each other!



# Subtask 3

$N = M$

- Therefore, in this subtask, we can simply match $D_1$ with $C_1$, $D_2$ with $C_2$, ..., and $D_N$ with $C_N$.
- You can simply calculate the maximum value of $|D_i - C_i|$ over $1 \le i \le N$.

| | | | | | |
|---|---|---|---|---|---|
| O( | N | ) | Expected Score | 39 | Cumulative | 51 |

# Subtask 4

The required answer does not exceed $30$

- This subtask is in fact close to full solution!

- Let’s say we have a procedure to answer the following quick enough:

  ***“ Is it possible that the answer does not exceed $x$? ”***
- Then, we can loop $x$ from $0$ to $30$ to find the minimal such $x$!

TWGSS OI Team

# Subtask 4

The required answer does not exceed $30$

***" Is it possible that the answer does not exceed $x$? "***

- How to check this quickly?
- We will use a greedy approach: for every deer, we will try to assign the leftmost remaining valid cracker to it.
  - Why? Optimally, there isn't any crossing.
  - Hence, the more left the cracker assigned to the leftmost deer, more crackers remain for the rest of the deers (which is better!)
- You can maintain this simply by a queue storing the cracker positions.

| O( **max(N, M) A** ) | Expected Score | **21** | Cumulative | 72 |
|---|---|---|---|---|

*with* maximum answer = **A**

# Full Solution

- We can improve the time complexity by using **binary search** on $x$!

- Initially, we set the lower bound $l = 0$ and the upper bound $r = 10^9$.
- Then, we can repeatedly shrink the range in half, by asking whether middle $m = \left\lfloor \frac{l+r}{2} \right\rfloor$ can be answer or not.

  If $m$ can be answer, set $r = m$,

  If $m$ cannot be answer, set $l = m + 1$.
- Repeat until $l = r$, then that value will be the answer!

| O( max(N, M) lgA ) | Expected Score | 100 | Cumulative | 100 |
|---|---|---|---|---|

*with* maximum answer = **A**

# Deer Shop

**TMF252**

# Problem Statement

- There are $N$ different stores selling deer crackers, with different quantities and prices.
- Given $K$ dollars, maximise the number of deer crackers Nokotan can buy.

# Subtask 1

$N = 1$

- There is only one store.
- Simple Maths problem!
- Solution left as exercise for *deer readers*.

| Expected Score | 9 |
| --- | --- |

| Cumulative | 9 |
| --- | --- |

# Subtask 2

$1 \le K \le 10^4$
$S_i = 10^4$

- The number of deer crackers at each store are basically infinity.
- Therefore, you can simply find the store that sells the cheapest deer crackers and buy all from that store.

| | | | | | |
|---|---|---|---|---|---|
| O( N ) | | Expected Score | 12 | Cumulative | 21 |

# Subtask 3

$1 \leq N \leq 5000$
$S_i = 1$

- How can we employ the greedy idea as discovered in Subtask 2?
- Simply, repeatedly find the cheapest shop, which the deer cracker hasn't been sold!

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| O( | **$N^2$** | ) | Expected Score | **40** | Cumulative | 52 |

TWGSS OI Team

# Subtask 4

$1 \leq S_i \leq 10$

- Let's say we have a list of prices of each stack of deer crackers.
- Obvious thing to do is to sort the deer crackers in ascending order of prices and buy the deer cracker one-by-one!

| O( **NS lg(NS)** ) | Expected Score | **62** | Cumulative | 78 |
|---|---|---|---|---|

# Full Solution

- We can slightly extend the previous solution – by consider a *store* as a unit, instead of each *stack of deer cracker*!
- This works as the total number of stores is bounded by $10^5$, but the total number of deer crackers can be as large as $10^9$!
- We can sort the stores (as pairs) in ascending order of prices and consider the prices one-by-one.

| O( | **N lgN** | ) |
| --- | --- | --- |

| Expected Score | **100** |
| --- | --- |

| Cumulative | 100 |
| --- | --- |

# Deer Friend

**TMF253**

# Problem Statement

- There are $N$ entities (deer + human).
- Each deer has a deer friend index of $0$.
- If an entity has a deer friend index of $K$, its friend can have a deer friend index of $K + 1$.
- If there is no "friend chain" connecting a human to a deer, then its deer friend index is $-1$.
- Each entity's deer friend index is minimised ($-1$ excluded).
- Find the deer friend index of all entities!

# Subtask 1

$N = 2$
$M = 1$

There are two entities, and they are friends...

Sanity check!

- If both are human, the answer is `-1 -1`
- If only one is human, the answer is either `0 1` or `1 0`
- If both are deer, the answer is `0 0`

Simple case-handling!

| Expected Score | 8 |
| --- | --- |

| Cumulative | 8 |
| --- | --- |

# Subtask 2

$D_1 = 0$
$D_i = 1$ for $2 \leq i \leq N$

Only entity $1$ is a human, and all other entities are deer.

- Therefore, the deer friend index of all deers are $0$.
- What remains to consider is entity $1$:
  - If entity $1$ is friend with some deer, then its deer friend index is $1$.
  - Otherwise, its deer friend index is $-1$.

- Also attempt-and-get-score subtask!

| Expected Score | 11 |
| --- | --- |

| Cumulative | 19 |
| --- | --- |

# Subtask 3

$M = N - 1$

$U_i = i, V_i = i + 1$ for $1 \leq i \leq M$

The friendships form a chain.

i.e., Entity $k$ friends with entity $k + 1$ for all $1 \leq k \leq N - 1$.

- Notice that what matters is the closest deer on the left and on the right for each entity.
- We can maintain it by considering entities one-by-one from left to right, and from right to left.

| Expected Score | 26 |
|---|---|

| Cumulative | 37 |
|---|---|

# Subtask 4

$D_1 = 1$
$D_i = 0$ for $2 \leq i \leq N$

Only entity $1$ is a deer, all others are human.

- Okay. This is a pretty standard task *(why would it appear in Mock???).*
- We can use **Breadth First Search (BFS)** to solve this task.
  - Basically we can maintain it using a deque (double-ended queue).
  - Firstly, push entity $1$ to the beginning of the deque.
  - While the deque is non-empty, we can do the followings:
    - Let's say the current front element is $x$.
      We push every friend of entity $x$ to the back if not yet pushed and update their deer friend indices.

| | | | | | |
|---|---|---|---|---|---|
| O( | **N + M** | ) | Expected Score | **35** | Cumulative 64 |

TWGSS OI Team

# Subtask 5

$1 \leq N \leq 5000$
$1 \leq M \leq 10^4$

- We can extend the idea in Subtask 4!
- Basically, we run BFS once from each deer.
- Then, we can take minimum over all possible deer friend indices.

| O( | **NM** | ) | Expected Score | **50** | Cumulative | 79 |
|---|---|---|---|---|---|---|

# Full Solution

- Indeed, we can still use BFS to solve this task (perhaps pretty well-known).
- At the beginning, we can push all the deers to the deque, and instead of running the BFS algorithms many times, we run it once only.
- And we are done!

| O( | **N + M** | ) | Expected Score | **100** | Cumulative | 100 |
|---|---|---|---|---|---|---|

TWGSS OI Team

# Deer Jane

**TMF254**

# Problem Statement

- Deer are running in circle, some clockwise and some anticlockwise.
- There are $N$ checkpoints and in a second, deer run from a checkpoint to an adjacent one.
- Handle $Q$ queries: count the number of deers in a range at a particular second.

# Subtask 1

$N = 2$
$Q = 1$

- Observation: after $N$ seconds, every deer back to the initial position.
- Therefore, what only matters is the parity of $T$!
  - If $T$ is even, then the position does not change.
  - If $T$ is odd, then every deer is at the other location.
- Then, you can think all the cases and solve them independently.
- Of course, it is better if you can rule out some “methods” in making your calculation accurate.

| Expected Score | 7 |
| --- | --- |

| Cumulative | 7 |
| --- | --- |

# Subtask 2

$Q = 1$
$T_1 = 0$

- The question becomes a partial sum problem!
- A trick is that instead of maintaining a partial sum of a length-$N$ array, you can copy the array once more and calculate the partial sum of a length-$2N$ array instead.
- Suppose partial sum array `S[R] = A[1] + A[2] + … + A[R]`, then
  - If `L < R`, then `count = S[R] - S[L - 1]`.
  - Otherwise, `count = S[R + N] - S[L - 1]`.

| Expected Score | 13 |
|---|---|
| Cumulative | 20 |

# Subtask 3

$Q = 1$
$A_i = 0$

- When there is only one query, you can simply compute the final status of the deers and employ Subtask 2 solution (aka partial sum).
- You can discover that for the deers moving in clockwise, they are shifted by `T % N` checkpoints.

| Expected Score | 21 |
| --- | --- |

| Cumulative | 41 |
| --- | --- |

# Subtask 4

$T_1 = T_2 = \cdots = T_Q$

- Similar to Subtask 3, you can also compute the final status after $T_j$ seconds and simply do partial sum on it.

| Expected Score | **60** |
|---|---|

| Cumulative | 60 |
|---|---|

# Subtask 5

$A_i = 0$

- Basically this is the full solution, as deers running clockwise and running anticlockwise are essentially symmetric.

- Instead of directly calculating the final status, you can observe that the answer is still a range of deers with some calculations.
- Therefore, it can still be done with the use of partial sum.

| Expected Score | 45 |
| --- | --- |

| Cumulative | 84 |
| --- | --- |

# Full Solution

- Simply handle anticlockwise deers correctly and you will get the remaining 16 points!
- Sounds easy, right?

| Expected Score | **100** |
| --- | --- |

| Cumulative | 100 |
| --- | --- |