

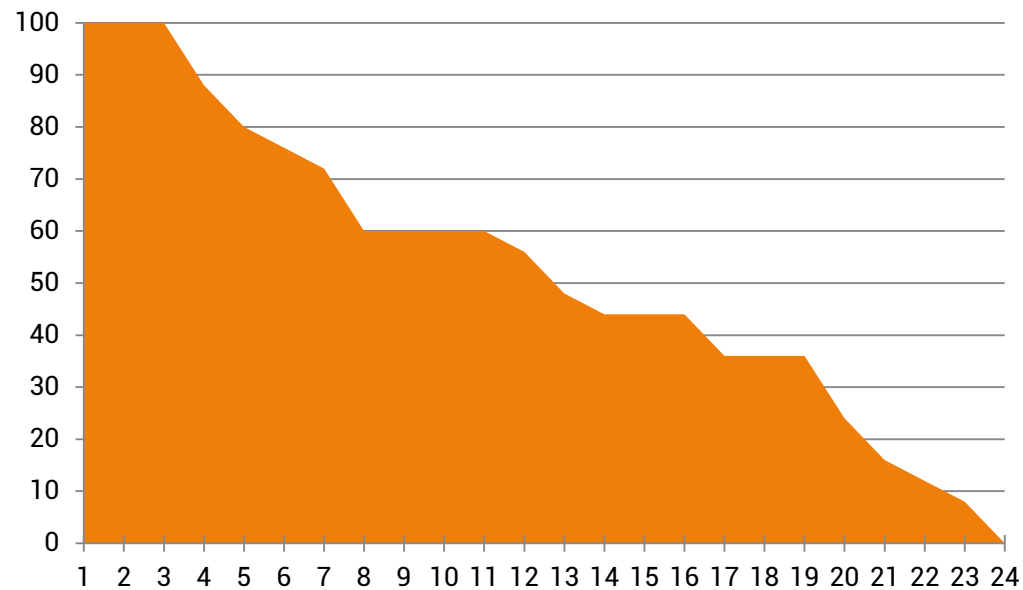


# **Mini Comp 0 Editorial**

**Tony Wong**

---

# Valentine Chocolates



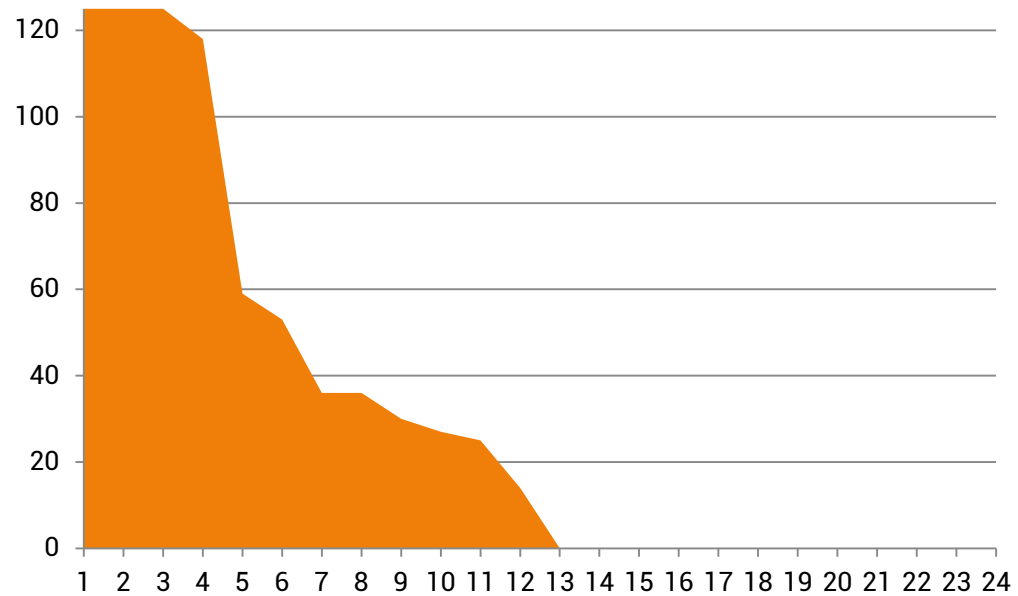
# Valentine Chocolates

- Rectangles: Find number of factors
- Interlaced: Find number of solutions  
 $xy + (x - 1)(y - 1) = N$
- Solution 1: Exhaust  $x$  and  $y$
- Expected score: 60

# Valentine Chocolates

- Solution 2: Exhaust  $x$ , check if  $y$  is integer
- $2xy - x - y + 1 = N$
- $y = (N - 1 + x) / (2x - 1)$
- Expected score: 100 assuming long long is used *properly*

# Valentine Dinner



# Valentine Dinner

- First convert all names to lowercase so that the comparison is case insensitive
- Subtask 2: For each item in the original menu, check to see which item in the Valentine's menu has the word

# Valentine Dinner

- Subtask 3:
- Just check each menu item
- Expected score: 125
- Make sure you check the words in order

- 2

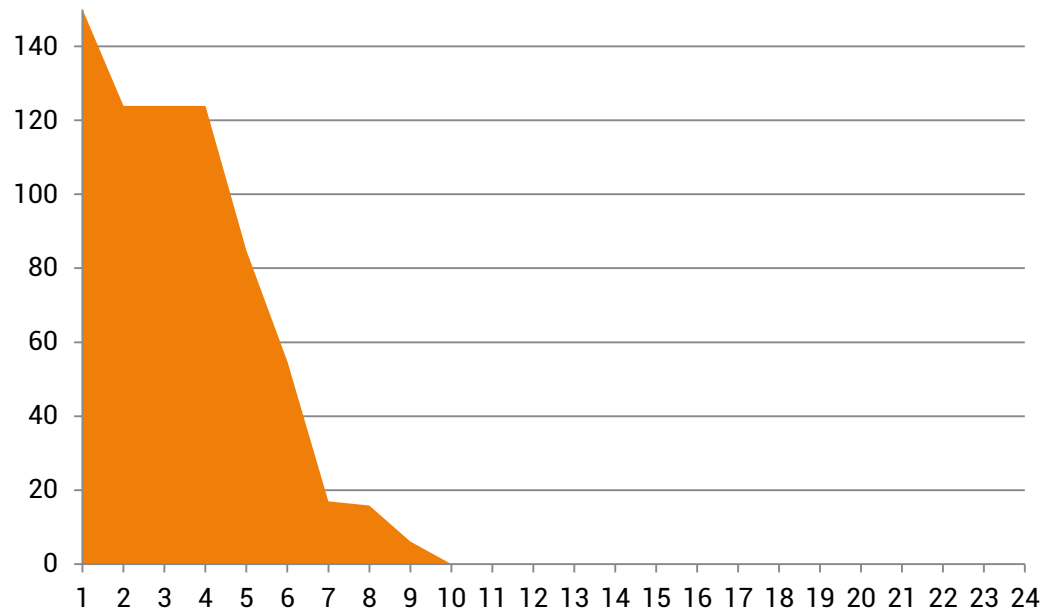
```
abc def
```

```
def abc
```

```
abc def ghi
```

```
def abc ghi
```

# Valentine Going Home





# Valentine Going Home

- Subtask 1:
- Check if it is possible to go from start to end by travelling to the right (positive  $x$ ) and upwards (positive  $y$ )
- Use a 2D array and process it like DP
- If possible, the output is  $x_N + y_N$

# Valentine Going Home

- Subtask 2:
- Assume the only possible route is 1, a, b, ..., N
- There is also exactly 1 possible to all intermediate stops a, b, ...
- Use DFS to find the distance from 1 to a, b, ...
- Output `dist[N]`

# Valentine Going Home

- Each bus route has at most 1 possible direction (0 if the distances are the same)
- They also point “outwards”
- Therefore it is a directed acyclic graph
- The problem is DAG longest path problem

# Valentine Going Home

- Subtask 3:
- Topological sort the bus stops
- For those nodes that the in-degree is 0
  - Set  $\text{dist}[i] = -\text{infinity}$ ; mark as processed
- Set  $\text{dist}[1] = 0$ ;
- Repeat:
  - Find a nodes that the all parent nodes are processed
  - $\text{dist of node} = \max(\text{dist}[\text{from}] + \text{euclidean dist})$
- Be careful with **long long**

# Valentine Going Home

- Alternative solution: dynamic programming
- Sort the bus stops in increasing distance
- $dp[i] = \max(dp[j] + \text{euclidean dist})$   
where there is a bus from  $j$  to  $i$