Competitive Programming and Mathematics Society

Launch Week contest debrief

CPMSoc

Table of contents

Prime Dice

CPMSOC

- Out of the numbers 1-6, only 2, 3 and 5 are prime. Thus, for each die there is a $3/6 = 1/2$ chance of rolling a prime number. Since we have 3 independent rolls, the odds of getting this outcome 3 times is $(1/2) \cdot (1/2) \cdot (1/2) = 1/8$.

Easy as A-B-C

- Using the AM-GM inequality, $\frac{a+2b+3c}{3} = \frac{1}{3} \geq \sqrt[3]{a \cdot 2b \cdot 3c} = \sqrt[3]{6}\sqrt[3]{abc}$, so $abc \leq \frac{1}{162}$.
- Note that equality is achieved when $a = 2b = 3c$, which requires $a + 2b + 3c = 3a = 1$
- Thus abc achieves its maximum of $\frac{1}{162}$ at $a = \frac{1}{3}, b = \frac{1}{6}, c = \frac{1}{9}$

Easy as A-B-C

- Using the AM-GM inequality, $\frac{a+2b+3c}{3} = \frac{1}{3} \geq \sqrt[3]{a \cdot 2b \cdot 3c} = \sqrt[3]{6}\sqrt[3]{abc}$, so $abc \leq \frac{1}{162}$.
- Note that equality is achieved when $a = 2b = 3c$, which requires $a + 2b + 3c = 3a = 1$
- Thus abc achieves its maximum of $\frac{1}{162}$ at $a = \frac{1}{3}, b = \frac{1}{6}, c = \frac{1}{9}$

Alternative solution (higher-dimensional version of using tangents to find maximum)

- Applying the constraint we wish to maximise $(1 - 2b - 3c)bc$. Since $a, b, c > 0$, at the boundary of our domain, $abc = 0$, so we need a positive stationary point. We take partial derivatives:

$$\frac{\partial}{\partial b}(1 - 2b - 3c)bc = (1 - 2b - 3c)c - 2bc = 0 \to 1 - 4b - 3c = 0$$
$$\frac{\partial}{\partial c}(1 - 2b - 3c)bc = (1 - 2b - 3c)b - 3bc = 0 \to 1 - 2b - 6c = 0$$

- Solving these equations simultaneously we get $abc = \frac{1}{162}$, which is > 0, so a maxima.

Inverted Integral

- Various solutions. For instance, we could graphically calculate the inverse function's area by using the integral of xe^x (exercise for reader :P).
- Or we can directly use substitution.

$$\int_{2e^2}^{3e^3} f(x)dx.x \to f^{-1}(u) = ue^u$$

- Note the bounds in terms of u become 2 and 3.

$$\int_2^3 ud(ue^u) = \int_2^3 u(e^u + ue^u)du$$

- We then integrate by parts, using $\int e^u = e^u$ and successively differentiating $u + u^2$:

$$\begin{aligned}\int_2^3 ue^u + u^2e^u du &= ((u+u^2)e^u - (1+2u)e^u + 2e^u)_2^3 \\ &= (e^u(u^2 - u + 1))_2^3 = 7e^3 - 3e^2.\end{aligned}$$

Chess 2

- The abomination moves a total of 20 moves right, and 20 moves up. This implies we have an even number of 2-right, 1-up moves and an even number of 2-up, 1-right moves. Since we end up moving the same total amount right as up, we also need an equal number of these two types of moves. We can now sum up all possibilities for a given number of straight diagonal moves. Suppose we have $2k$ 2-right moves. We then move diagonally based upon $\begin{bmatrix}20\\20\end{bmatrix} - 2k\begin{bmatrix}2\\1\end{bmatrix} - 2k\begin{bmatrix}1\\2\end{bmatrix} = \begin{bmatrix}20-6k\\20-6k\end{bmatrix}$ gives $10-3k$ diagonal moves. We have a total $10-3k+2k+2k = 10+k$ moves, choosing $4k$ to be non-diagonal. These $4k$ moves can be performed in $\binom{10+k}{4k}$ different positions, and rearranged among themselves $\frac{(4k)!}{(2k)!^2}$ ways. We make the sum

$$\binom{10}{0}\frac{0!}{0!^2} + \binom{11}{4}\frac{4!}{2!^2} + \binom{12}{8}\frac{8!}{4!^2} + \binom{13}{12}\frac{12!}{6!^2}$$

- This gives 48643 ways.

Painting

- Observation: Artworks are purely distinguishable by which seats are coloured red.
- Change order of summation: add scores for each seat instead of each configuration (i.e. in how many artworks is it coloured).
- A given seat is coloured if at least one of it or its two adjacent seats are coloured (7 possibilities), leaving 2^{n-3} colourings for the other seats.
- So there are $7 \times 2^{n-3}$ artworks where that seat is coloured.
- Add this for all n seats: $7n \times 2^{n-3}$

Find a Distance

CPMSOC

- TL;DR write the area of region AXB in two different ways: half that of the quadrant, and in terms of OX.
- Region $AXD =$ Sector $AOD - \triangle AOD = \frac{\pi}{12} - \frac{1}{2}\sin 30^\circ AO \cdot OX = \frac{\pi}{12} - \frac{OX}{4}$
- Quadrant $BOD = \frac{\pi}{4}$
- $\triangle BOX = \frac{1}{2}BO \cdot OX = \frac{1}{2} \cdot OX$
- Region $AXB =$ Quadrant BOD - Region AXD - $\triangle BOX = \frac{\pi}{8}$
- Therefore:

$$\frac{\pi}{4} - \frac{\pi}{12} + \frac{OX}{4} - \frac{OX}{2} = \frac{\pi}{8}$$
$$\frac{OX}{4} = \frac{\pi}{24}$$
$$OX = \frac{\pi}{6}$$

Locate a Function - Part 1

- To satisfy the condition $f(u+v) = f(u) + f(v)$ the function should be of the form $x_1 \cdot c_1 + x_2 \cdot c_2 + \cdots + c_n$
- To satisfy the injectivity condition, there are a number of possible choices of c_i. Some good ones are listed below:
  - $c_i = \ln p_i$ where p_i is the ith prime
  - $c_i = \sqrt{p_i}$
  - $c_i = \sqrt[i]{2}$

Locate a Function - Part 2

- Realise all possible solutions to the previous are of a certain form: $x_1 \cdot c_1 + x_2 \cdot c_2 + \cdots$
- Why? $f(u+v) = f(u) + f(v)$ is Cauchy's functional equation, which is linear over $\mathbb{Z}$ and (as it happens) $\mathbb{Z}^\infty$ (i.e. $zf(x) = f(xz)$. This is not true for a real domain!)
- From $zf(x) = f(xz)$, we notice that:

$$f(x) = f\left(x_1 \begin{pmatrix} 1 \\ 0 \\ \vdots \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 1 \\ \vdots \end{pmatrix} + \cdots \right) = x_1 f\left(\begin{pmatrix} 1 \\ 0 \\ \vdots \end{pmatrix}\right) + x_2 f\left(\begin{pmatrix} 0 \\ 1 \\ \vdots \end{pmatrix}\right) + \cdots$$
$$= x_1 \cdot c_1 + x_2 \cdot c_2 + \cdots$$

- Choose the integers $x_1, x_2, \cdots$ such that $x_i c_i \geq 1$ (note that c_i is non-zero otherwise injectivity is violated).
- $f(x)$ cannot exist since $x_1 \cdot c_1 + \cdots$ is a divergent series, so f cannot satisfy all prescribed conditions

Battleship

- Solution: Find subarray of length k with largest sum (contiguous k elements in the array)
- Time complexity (within time limit):
  - $O(N)$ - prefix sum/2-pointer/Sliding Window
  - $O(N \log N)$

Bombing Trees

- First observation: optimal to pick a length of k where the most monkey-bombs have been dropped (so apart of the bounds 0 and 10^9 where bombs have not been dropped, the rest of the infinite grid can be ignored)
- Solution: find subarray from 0 to 10^9 of length k with most bombs dropped inside

Bombing Trees

- First observation: optimal to pick a length of k where the most monkey-bombs have been dropped (so apart of the bounds 0 and 10^9 where bombs have not been dropped, the rest of the infinite grid can be ignored)
- Solution: find subarray from 0 to 10^9 of length k with most bombs dropped inside
  - Sliding window: $O(N)$ solution (with $O(N \log N)$ sorting pre-computation)
  - Check windows where all bombs are within k distance of each other

Candy Store

- Solution:
  - Brute force - check every permutation of candy in each bag, has $O(k^n)$ time complexity (works for first 2 subtasks)
  - Modified knapsack DP algorithm - check weight of each bag (has $O(n^4w^3)$ time complexity -> can be optimised to $O(n^3w^2)$)

Alice and Hamster

- Observations:
  - The connections form a graph
  - Each checkpoint is only connected to a single checkpoint with a smaller number
  - This means the checkpoints form a tree!
- Solution:
  - $n^2 log(a)$ Brute force - for each pair of special checkpoints, find the Lowest Common Ancestor and thus the path length.
  - $nlog(a)$ build the tree with leaves as special nodes. For each node, calculate the depth a of the deepest node in its subtree . For each node, the longest path peaking at it = sum of the two largest a's of it's children. The longest path is the maximum of these.