30 August 2023 14:42

Matrix logic

Distinct numbers are arranged in an $m \times n$ rectangular table with m *rows* and n *columns* so that in each row the numbers are in increasing order (left to right), and in each column the numbers are in increasing order (top to bottom). Such a table is called a *sorted* table and each location of the table containing a number is called a *cell*. Two examples of sorted tables with 3 rows and 4 columns (and thus $3 \times 4 = 12$ cells) are shown below.

3	12	33	64
15	26	37	78
19	40	51	92

5	22	53	68
18	36	67	78
19	45	81	92

We index the cells of the table with a pair of integers (i, j), with the top-left corner being $(1, 1)$ and the bottom-right corner being (m, n). Observe that the smallest entry in a sorted table can only occur in cell $(1, 1)$; however, note that the second smallest entry can appear either in cell $(1, 2)$, as in the first example above, or in cell $(2, 1)$ as in the second example above.

(i) (a) Assuming that $m, n \geqslant 3$, where in an $m \times n$ sorted table can the third-smallest entry appear?

(b) For any $k \geqslant 4$ satisfying $m, n \geqslant k$, where in an $m \times n$ sorted table can the k^{th} smallest entry appear? Justify your answer.

(ii) Given an $m \times n$ sorted table, consider the problem of determining whether a particular number y appears in the table. Outline a procedure that inspects at most $m + n - 1$ cells in the table, and that correctly determines whether or not y appears in the table. Briefly justify why your procedure terminates correctly in no more than $m + n - 1$ steps.
[Hint: As the first step, consider inspecting the top-right cell.]

(iii) Consider an $m \times n$ table, say A, which might not be sorted; an example is shown below. Obtain table B from A by re-arranging the entries in each row so that they are in sorted order. Then obtain table C from B by re-arranging the entries in each column so that they are in sorted order. Fill in tables B and C here:

A:

33	92	46	24
25	26	37	8
49	40	81	22

$\rightarrow$ B:

$\rightarrow$ C:

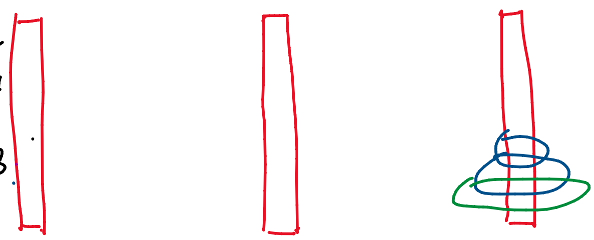
(iv) Show that for *any* $m \times n$ table A, performing the two operations from part (iii) results in a sorted table C.

	rings	moves
t_1	1	1
t_2	2	3
t_3	3	7
t_4	4	15

2, 4, 8

$t_n = ?$

t_{50}



A B C.

Transfer all the discs from pole A to pole C using the following conditions.

① you cannot move more than one disc at a time

② you can use pole B.

③ at no point in time ...

a) In how many moves can you achieve this? 15

a) In how many moves can you achieve this? 15

b) If there are 50 discs how many moves are needed.

③ at no point in time should a bigger disc be placed on a smaller disc.

$t_1 = 1$

$t_2 = 3$

$t_3 = 7$

$t_4 = 15$

$$\boxed{t_n = 2^n - 1}$$

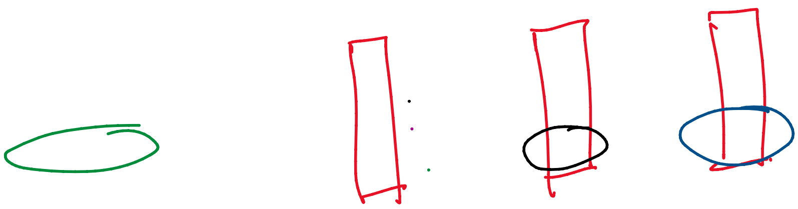
$$\cancel{t_2} - t_1 = 2^1$$
$$\cancel{t_3} - \cancel{t_2} = 2^2$$
$$\cancel{t_4} - \cancel{t_3} = 2^3$$
$$\vdots$$
$$t_n - \cancel{t_{n-1}} = 2^{n-1}$$

$$t_n - t_1 = 2^n - 2$$
$$t_n - 1 = 2^n - 2$$

$$S_n = a \cdot \frac{r^n - 1}{r - 1}$$

$$S_{n-1} = a \cdot \frac{r^{n-1} - 1}{r - 1}$$
$$= 2 \cdot \frac{2^{n-1} - 1}{2 - 1}$$
$$= 2(2^{n-1} - 1)$$
$$= 2^n - 2$$

ⓒ In a stack of 10 discs how many moves will be required to move the 7th disc from the top?



Tower of Hanoi

no of discs	Reqd move
1	$1 = 2^{1-1}$
2	$2 = 2^{2-1}$
3	$4 = 2^{3-1}$
4	$8 = 2^{4-1}$

7th disc → $2^{7-1} = 2^6 = 64$th move.

Row →

7	19	11	8	9
17	16	18	6	14
5	2	20	15	4
10	3	13	12	1

1–20 are placed at random in the boxes (no repetition)

In how many minimum moves can you arrange the nos in ascending order subject to the following conditions.

① you can swap 2 of them at a time

① you can swap 2 of them at a time
② While swapping row wise you can swap 2 even nos.
③ while swapping column wise you can swap a even with an odd.

C O N C A T E N A T E.

Arrange in ascending order by swapping.
What is the minimum no of swaps? and the maximum no of swaps.

Bubble sort

①.
②.
③.
④
⑤.

A A C C E E N | N O T | T.

A A C C E E N | N. O T | T.

Jail

□ □ □ □ □ □ □ □ □ □
1 2 3 4 5 6 7 8 9 10.

Assumption:
Intruder will act logically to free all the prisoners in the shortest time.

① He can open maximum 2 doors at a time.
② The doors opened must be in even-even or odd-odd pairs.
③ A sentry closes the first open door that he sees.
In 1 min the intruder can open 2 doors and in the next 1 min the sentry can close 1 door.
The prisoners can escape only when all doors are opened.
How long will it take to free the prisoners.
④ Both the intruder and the sentry takes 1 min to go from one door to the next.

+91 8105735888.

sukamal_chat@yahoo.com.

sukamal_char@g