

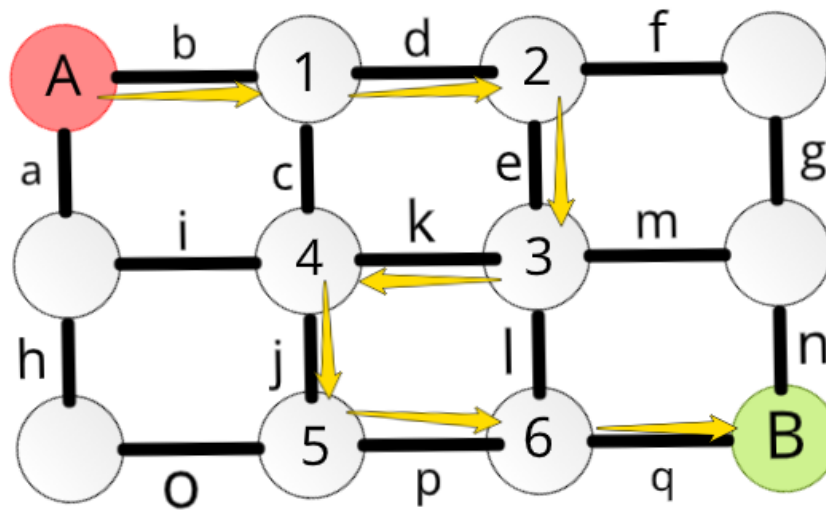
# andpath

You are given an  $R * C$  grid of cities and you are initially at  $0, 0$  and have to reach  $R - 1, C - 1$  through other cities. Moving from one city to adjacent one fetches you some value. You can move in any 4 directions (Right, Left, Up, Down). You cannot visit any city twice.

Formally, you should consider all self-avoiding walks from  $0, 0$  to  $R - 1, C - 1$

Suppose you start at  $0, 0$  and reach  $R - 1, C - 1$  and the path traced by you has values  $p_1, p_2, p_3 \dots p_x$ . The value of that path is defined as  $(p_1 \wedge p_2 \wedge p_3 \wedge \dots \wedge p_x)$  where ' $\wedge$ ' is the bitwise-and operator. At the city  $R - 1, C - 1$  you can exchange the path-value to gold coins.

What is the maximum path-value you can achieve?



In the above example, the path is  $A \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow B$  and the value of this path is  $(b \wedge d \wedge e \wedge k \wedge j \wedge p \wedge q)$

The input for the above image would be

```
1
3 4
a b c d e f g h i j k l m n o p q
```

Note: Use fast io as input files are as large as 15MB.

## Input Format

The first line contains  $T$ , the number of testcases.

The first line of each testcase has two integers,  $R$  and  $C$ ; Number of rows and columns in that order.

The second line of each testcase has  $(2 * R * C - R - C)$  integers which are the values.

## Constraints

$1 \leq T \leq 500$

$1 \leq R \cdot C \leq 10^5$

$1 \leq value \leq 2^{63}$

The total number of integers in a file will not exceed  $10^6$

Output Format

For each testcase print a single integer containing the maximum path value from  $0,0$  to  $R-1,C-1$

Sample Input 0

```
4
3 4
4128 12522 14530 15638 9339 29359 5978 5273 20976 32066 7148 19756 19077 8180 21808 13318 18852
5 2
25225 20778 15867 19028 12051 12412 12238 2161 17044 31230 4528 5887 8535
3 2
5260 25644 18064 15434 14104 27488 18037
5 5
23421 5399 21255 7560 31880 21973 12477 15928 1063 17763 7677 14449 24384 11173 21794 27927 4076 6594 9592
14338 2965 4850 1750 8156 12355 7845 25639 5999 24258 7606 2385 1418 12753 12123 8120 31408 13801 4004 1494
27894
```

Sample Output 0

```
4096
0
16384
4096
```