Grind 3

29 April 2025 17:17

A. Sum in Binary Tree

time limit per test: 1 second

memory limit per test: 256 megabytes

Vanya really likes math. One day when he was solving another math problem, he came up with an interesting tree. This tree is built as follows.

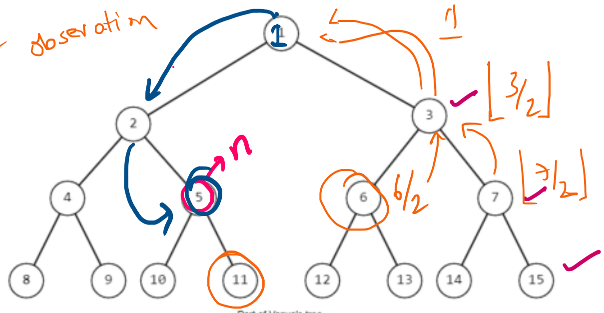
Initially, the tree has only one vertex with the number 1 — the root of the tree. Then, Vanya adds two children to it, assigning them consecutive numbers — 2 and 3, respectively. After that, he will add children to the vertices in increasing order of their numbers, starting from 2, assigning their children the minimum unused indices. As a result, Vanya will have an infinite tree with the root in the vertex 1, where each vertex will have exactly two children, and the vertex numbers will be arranged sequentially by layers.

→ Got ✓

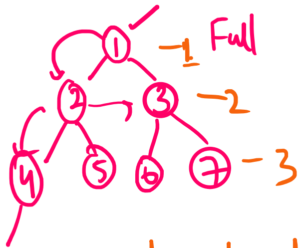
→ K →

BT → X

sight observation



Part of Vanya's tree.



max value of node at level 1 ✓

1 , Level - 3

$\sum n \to n/2 \to n/4 \cdots 1$

Example

input

```
6
3
10
37
1
10000000000000000
15
```

output

```
4
18
71
1
19999999999999980
26
```

$1, 3, 7, 15$ → observe

$2^1-1, 2^2-1, 2^3-1, 2^4-1$

order of 2 ✓

↓

for inc / decreasing

$\boxed{n + n/2 + n/4 + n/8 \cdots}$ ✓

C. Increasing Sequence

→ question

↓

time limit per test: 1 second

memory limit per test: 64 megabytes

A sequence $a_0, a_1, \ldots, a_{t-1}$ is called increasing if $a_{i-1} < a_i$ for each i: $0 < i < t$.

You are given a sequence $b_0, b_1, \ldots, b_{n-1}$ and a positive integer d. In each move you may choose one element of the given sequence and add d to it. What is the least number of moves required to make the given sequence increasing?

Input

The first line of the input contains two integer numbers n and d ($2 \le n \le 2000$, $1 \le d \le 10^6$). The second line contains space separated sequence $b_0, b_1, \ldots, b_{n-1}$ ($1 \le b_i \le 10^6$).

Output

Output the minimal number of moves needed to make the sequence increasing.

Examples

input

```
4 2
1 3 3 2
```

output

```
3
```

[1, 3, 3, 2] → Strictly increasing

[1, 3, 3, 3, 3] → X

1 = 1 = -

$a_i < a_i$

[1, 3, 3, 3, 3] → x

$a_{i-1} < a_i$

1 operation, d - is given □↑ by d in 1 operation

[1, 3, 3, 3, 3] → [1, 3, 5, 3, 3] → Strictly increasing by d↑

one operation ✓

in minimum operation

[1, 3, 3, 2]

d = 2

adj elements

two pointers ✓

[1, 3, 3, 2]

yes ✓

[1, 3, 3, 2] → if equal or if cur element is less than prev element,

d

ops - ~~1~~ ~~2~~ 3

ans

[1, 3, 5, 2]

> prev

[1, 3, 5, 2]

formula

figure

[1, 3, 5, 4] → [1, 3, 5, 6]

E. Sasha and Array Coloring

time limit per test: 1 second
memory limit per test: 256 megabytes

Sasha found an array a consisting of n integers and asked you to paint elements.

You have to paint each element of the array. You can use as many colors as you want, but each element should be painted into exactly one color, and for each color, there should be at least one element of that color.

The *cost* of one color is the value of $\max(S) - \min(S)$, where S is the sequence of elements of that color. The *cost* of the whole coloring is the **sum** of costs over all colors.

For example, suppose you have an array $a = [1, 5, 6, 3, 4]$, and you painted its elements into two colors as follows: elements on positions 1, 2 and 5 have color 1; elements on positions 3 and 4 have color 2. Then:

- the cost of the color 1 is $\max([1, 5, 4]) - \min([1, 5, 4]) = 5 - 1 = 4$;
- the cost of the color 2 is $\max([6, 3]) - \min([6, 3]) = 6 - 3 = 3$;
- the total cost of the coloring is 7.

For the given array a, you have to calculate the **maximum** possible cost of the coloring.

Input

The first line contains one integer t ($1 \le t \le 1000$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \le n \le 50$) — length of a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 50$) — array a.

Output

For each test case output the maximum possible cost of the coloring.

Example

input

```
6
5
1 5 6 3 4
1
5
4
```

→ a ✓

[1, 2, 3, 4] ↗ ① ✓

→ Among ~~all~~ some colours max - min

[1, 2], [3, 4]

2 - 1 = 1
+
4 - 3 = 1

2

4 ✓ 1

3 ✓ 2

Example

input

```
6
5
1 5 6 3 4
1
5
4
1 6 3 9
6
1 13 9 3 7 2
4
2 2 2 2
5
4 5 2 2 3
```

output

```
7
0
11
23
0
5
```

Note

In the first example one of the optimal coloring is $[1, 5, 6, 3, 4]$. The answer is $(5-1)+(6-3)=7$.

In the second example, the only possible coloring is $[5]$, for which the answer is $5-5=0$.

In the third example, the optimal coloring is $[1, 6, 3, 9]$, the answer is $(9-1)+(6-3)=11$.

$2-1=1$
$+$
$4-3=1$
2

$\boxed{3 \checkmark \quad 2}$

$[1\ 2\ 3\ 4]$

$4-1=3$
$+$
$3-2=1$
4

K

Many colours are possible

$\boxed{\text{Greedy}}$

$[1, 2, 3, 4] \rightarrow 4-1=3 < 4$

>1 <0 >1

observation → split into as many colours as possible.

1) Sort the array

2) Two pointers

$[x, x+1, x+2, x+3, x+4]$

Last E – First E

3) Sum → Running → Ans

B)

Note

In the second example, there is a permutation such that in each cyclic shift there is a fixed point (highlighted in dark red):

1	2	3	4	5
4	1	3	5	2

1	2	3	4	5
2	4	1	3	5

1	2	3	4	5
5	2	4	1	3

1	2	3	4	5
3	5	2	4	1

1	2	3	4	5
1	3	5	2	4

The first line contains the element numbers, and the second line contains all the shifts of the desired permutation.

Easy

Logical

$\boxed{\text{Greedy problems}}$ → build that logic → ps

$n=2$ } $1-2$ → $1,2$ or $2,1$

$n=3$ } $1-3$

n = 3 } 1 – 3

1,2,3 or 3,2,1 or 1,3,2 . . .

permutations ✓

n numbers → i == num

i = 1 2 3 4 5

4 1 3 5 2

2 4 1 3 5

5 2 4 1 3

3 5 2 4 1

1 → n

[1,2] ✓

n – 1 → 10^5

1

i

{ 1 [1] } ✓

1 ~~2~~ 1

2 ~~1~~ 2

n=2

n=3 n=4

permutation

n=? ✗

idea → n = even ✗ ✓

impossible

1 2 3 4 = n is even

1 3 ~~4~~ 2 4 → not ✓

n – is even ✗

3 2 4 1

✗ wrong ✗

–1

–1 n is even

1 3 2 4

n = 3 ✓

1 2 3

all can

1 2 3

1 3 2 } possible ✓

1 3 5 2 4

n=5

n=4 ✗

1 2 3 4 5

1 2 3
1 3 2 → 1
2 1 3
3 2 1 →

n=4 ✗

n=5 -

n=5

1 2 3 4 5
1 3 5 2 4
2
4 1 3 5 2
2 4 1 3 5

starting off with all odds

valid
{ 1 3 2 → n=3
{ 1 3 5 2 4 → n=5

n=7

1 2 3 4 5 6 7
1 3 5 7 2 4 6
6 1 3 5 7 2 4 ✗

print all odds possible
↓
printing all evens possible

* Many
↓
opp 7 6 5 4 3 2 1

} good solution

→ prove
idea → ps

D. Deadly Laser

time limit per test: 2 seconds
memory limit per test: 256 megabytes

The robot is placed in the top left corner of a grid, consisting of n rows and m columns, in a cell $(1, 1)$.

In one step, it can move into a cell, adjacent by a side to the current one:

- $(x, y) \to (x, y+1)$;
- $(x, y) \to (x+1, y)$;
- $(x, y) \to (x, y-1)$;
- $(x, y) \to (x-1, y)$.

The robot can't move outside the grid.

The cell (s_x, s_y) contains a deadly laser. If the robot comes into some cell that has distance less than or equal to d to the laser, it gets evaporated. The distance between two cells (x_1, y_1) and (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$.

$|x_1 - x_2| + |y_1 - y_2|$

Print the smallest number of steps that the robot can take to reach the cell (n, m) without getting evaporated or moving outside the grid. If it's not possible to reach the cell (n, m), print -1.

The laser is neither in the starting cell, nor in the ending cell. The starting cell always has distance greater than d to the laser.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of testcases.

The only line of each testcase contains five integers n, m, s_x, s_y, d ($2 \le \boxed{n, m \le 1000}$; $1 \le s_x \le n$; $1 \le s_y \le m$; $0 \le d \le n+m$) — the size of the grid, the cell that contains the laser and the evaporating distance of the laser.

The laser is neither in the starting cell, nor in the ending cell ($(s_x, s_y) \ne (1, 1)$ and $(s_x, s_y) \ne (n, m)$). The starting cell $(1, 1)$ always has distance greater than d to the laser ($|s_x - 1| + |s_y - 1| > d$).

Output

For each testcase, print a single integer. If it's possible to reach the cell (n, m) from $(1, 1)$ without getting evaporated or moving outside the grid, then print the smallest amount of steps it can take the robot to reach it. Otherwise, print -1.

Example

input

```
3
2 3 1 3 0
2 3 1 3 1
5 5 3 4 1
```

output

```
3
-1
8
```

$|1-1| + |2-1| \Rightarrow 0 + 1$

3x3

(1,1) (1,2) (1,3) → 1

n x m

(3,3)

$2^n \to 2^{1000}$ ← TLE

n, m ← 1000

N^2 → 1000 x 1000

10^6

s → d

1 0 0
0 0 0
0 0 0

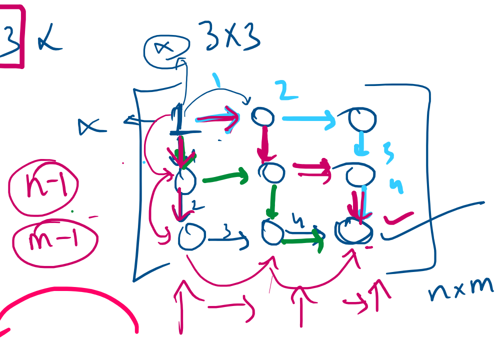
1

if then

$\boxed{n=2, m=3}$ ✗

3x3

$\boxed{n=2, m=3}$



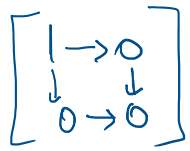
nxm → 3x3

Dry

minimum

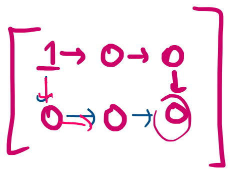
4 steps
4 steps
4 steps
4 steps

4steps → 3x3 ✓



2steps → 2x2

2x3 →



3steps
3 steps

formula →

n = 1000 ✓

3x3 ⇒ 9
2x2 → 4 ✓
2x2 → 1 x 1 → 1

$\boxed{3 \times 3 \rightarrow 4 \text{steps}}$
2 x 2 → 2steps
2 x 3 → 3 Steps

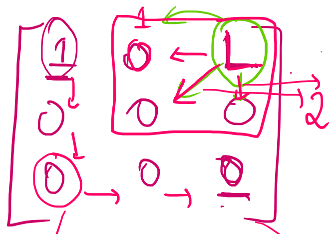
none ✓

Formula → n-1 + m-1 ⇒ n+m-2

3+3 - 2 = 4 ✓
2+2 - 2 ⇒ 2
2+3 - 2 ⇒ 3 ✓

formula → done ✓

$\boxed{n+m-2}$ → minimum steps ✓



→ L → 1 or nxm all

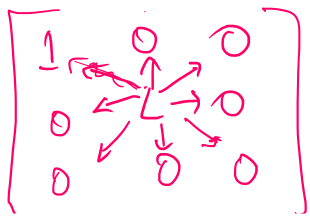
1 → L → d → d

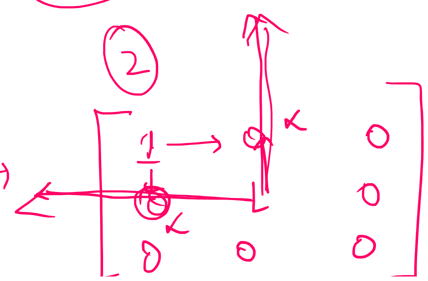
d = 2 ✓

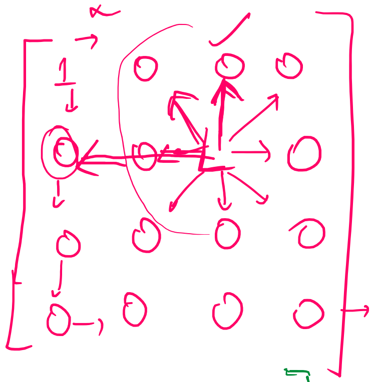
|2-1| + |3-2|
↳ 2

[1, 3]
[2, 2]

solving → n+m-2

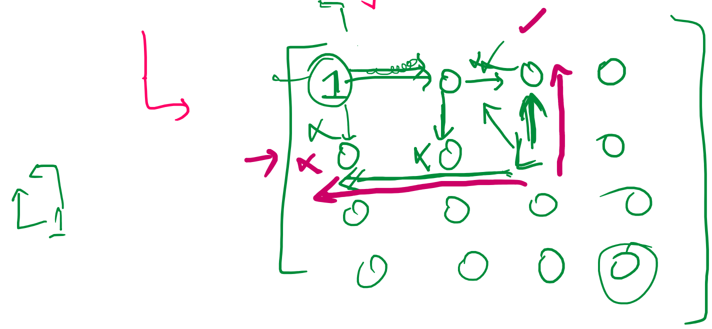






$d = 2,\ d = 3$

$n+m-2$ ✓



1 blocked

shoot both

-1

$n+m-2$