Beirut Arab University
Faculty of Science
Mathematics & Computer Science Department

Course: CMPS441 - Fundamentals of Algorithms
Semester: Fall 2024-2025
Lab 3

Part I:

Bubble Sort:

Bubble sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Your task is to implemets the below algorithm

0	1	2	3	4	5	6	7	*Process*
27	49	35	37	15	75	63	65	Original array
27	35	49	37	15	75	63	65	49, 35 interchanged
27	35	37	49	15	75	63	65	49, 37 interchanged
27	35	37	15	49	75	63	65	49, 15 interchanged
27	35	37	15	49	63	75	65	75, 63 interchanged
27	35	37	15	49	63	65	75	75, 65 interchanged
27	35	15	37	49	63	65	75	37, 15 interchanged
27	15	35	37	49	63	65	75	35, 15 interchanged
15	27	35	37	49	63	65	75	27, 15 interchanged

Pseudocode:

```
function bubbleSort(array)
  for i = 0 to array.length - 1
    for j = array.length - 1 down to i + 1
      if array[j] < array[j - 1]
        swap(array[j], array[j - 1])
```

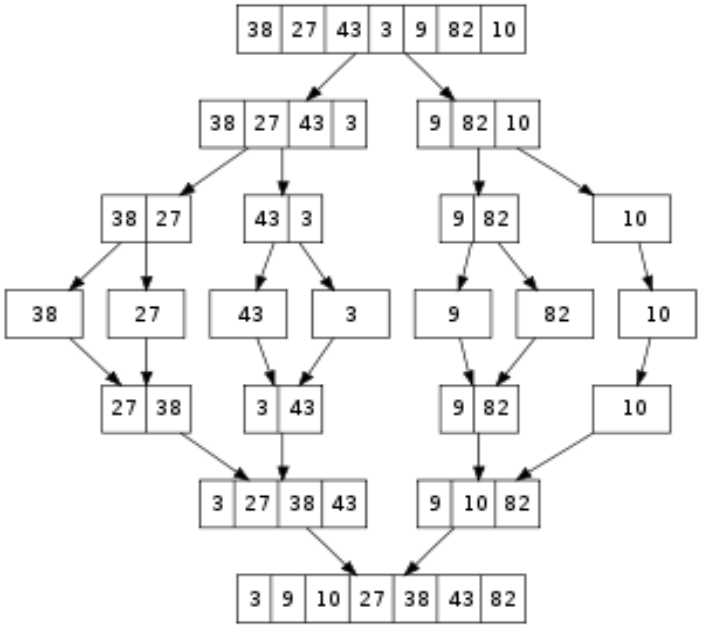
Part II:

1. Write a program which implements the merge sort algorithm presented as follows:

```
MERGE(A, p, q, r)
 1  n1 ← q - p + 1
 2  n2 ← r - q
 3  create arrays L[1 □ n1 + 1] and R[1 □ n2 + 1]
 4  for i ← 0 to n1
 5       do L[i] ← A[p + i]
 6  for j ← 0 to n2
 7       do R[j] ← A[q + 1 + j]
 8  L[n1] ← ∞
 9  R[n2] ← ∞
10  i ← 0
11  j ← 0
12  for k ← p  down to r
13       do if L[i] ≤ R[j]
14             then A[k] ← L[i]
15                  i ← i + 1
16             else A[k] ← R[j]
17                  j ← j + 1
```

```
MERGE-SORT(A, p, r)
1 if p < r
2   then q ← ⌊(p + r)/2⌋
3        MERGE-SORT(A, p, q)
4        MERGE-SORT(A, q + 1, r)
5        MERGE(A, p, q, r)
```

Example:



Part III:

Solve the following Codeforces problem using the merge sort algorithm. Submit your solution and make sure to get the "Accepted" verdict.
Problem link: https://codeforces.com/contest/1165/problem/B

Polycarp wants to train before another programming competition. During the first day of his training he should solve exactly 1 problem, during the second day — exactly 2 problems, during the third day — exactly 3 problems, and so on. During the k-th day he should solve k problems.

Polycarp has a list of n contests, the i-th contest consists of a_i problems. During each day Polycarp has to choose **exactly one** of the contests he didn't solve yet and solve it. He solves **exactly k problems from this contest**. Other problems are discarded from it. If there are no contests consisting of at least k problems that Polycarp didn't solve yet during the k-th day, then Polycarp stops his training. How many days Polycarp can train if he chooses the contests optimally?

Input

The first line of the input contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the number of contests.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2 \cdot 10^5$) — the number of problems in the i-th contest.

Output

Print one integer — the maximum number of days Polycarp can train if he chooses the contests optimally.

Examples

input

```
4
3 1 4 1
```

output

```
3
```

input

```
3
1 1 1
```

output

```
1
```

input

```
5
1 1 1 2 2
```

output

```
2
```