Amazon

1. 1710. Maximum Units on a Truck -- Easy/ Fill the truck/ load_cargo
2. Find the Highest Profit
3. Items in Containers / Items in compartment +++
4. 1010. Pairs of Songs With Total Durations Divisible by 60 -- Medium
5. Optimal Utilization ++ / Prime Air time++
6. 1167. Minimum Cost to Connect Sticks
7. 3. Longest Substring Without Repeating Characters -- Medium
8. 1792. Maximum Average Pass Ratio -- Medium / Five Star Seller priority queue based on how much increase the class could be
9. 974.Subaray Sum Divisible by K / Music Pairs / Pairs of Songs With Total Durations Divisible by 60 -- Medium $(a+b)\%60 = (a\%60 + b\%60)\%60$, **But watch out the negative number** `A[i] = sum % K >= 0 ? sum % K : sum % K + K;`
10. 146. LRU Cache -- Medium
11. 1629. Slowest Key -- Easy
12. 692. Top K Frequent Words -- medium map + priorityqueue + custom node
13. Cost Evaluation / CloudFront Caching ++++++

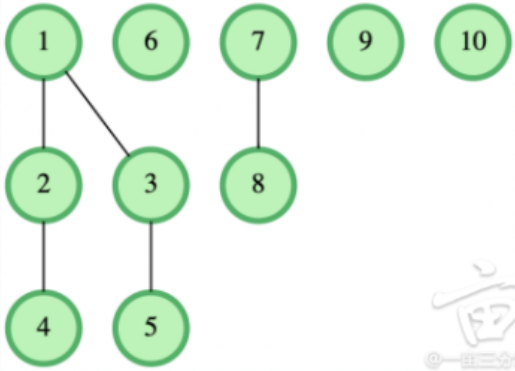
1. Cloudfront Caching

AWS CloudFront wants to build an algorithm to measure the efficiency of its caching network. The network is represented as a number of nodes and a list of connected pairs. The efficiency of this network can be estimated by first summing the cost of each isolated set of nodes where each individual node has a cost of 1. To account for the increase in efficiency as more nodes are connected, update the cost of each isolated set to be the ceiling of the square root of the original cost and return the final sum of all costs.

Example

n = 10 nodes

edges = [[1 2], [1 3], [2 4], [3 5], [7 8]]



There are *2* isolated sets with more than one node, *{1, 2, 3, 4, 5}* and *{7, 8}*. The ceilings of their square roots are $5^{1/2} \approx 2.236$ and *ceil(2.236) = 3*, $2^{1/2} \approx 1.414$ and *ceil(1.414) = 2*. The other three isolated nodes are separate and the square root of their weights is $1^{1/2} = 1$ respectively. The sum is *3 + 2 + (3 * 1) = 8*.

Function Description

Complete the function *connectedSum* in the editor below.

connectedSum has the following parameter(s):

int n: the number of nodes

str edges[m]: an array of strings that consist of a space-separated integer pair that denotes two connected nodes, *p* and *q*

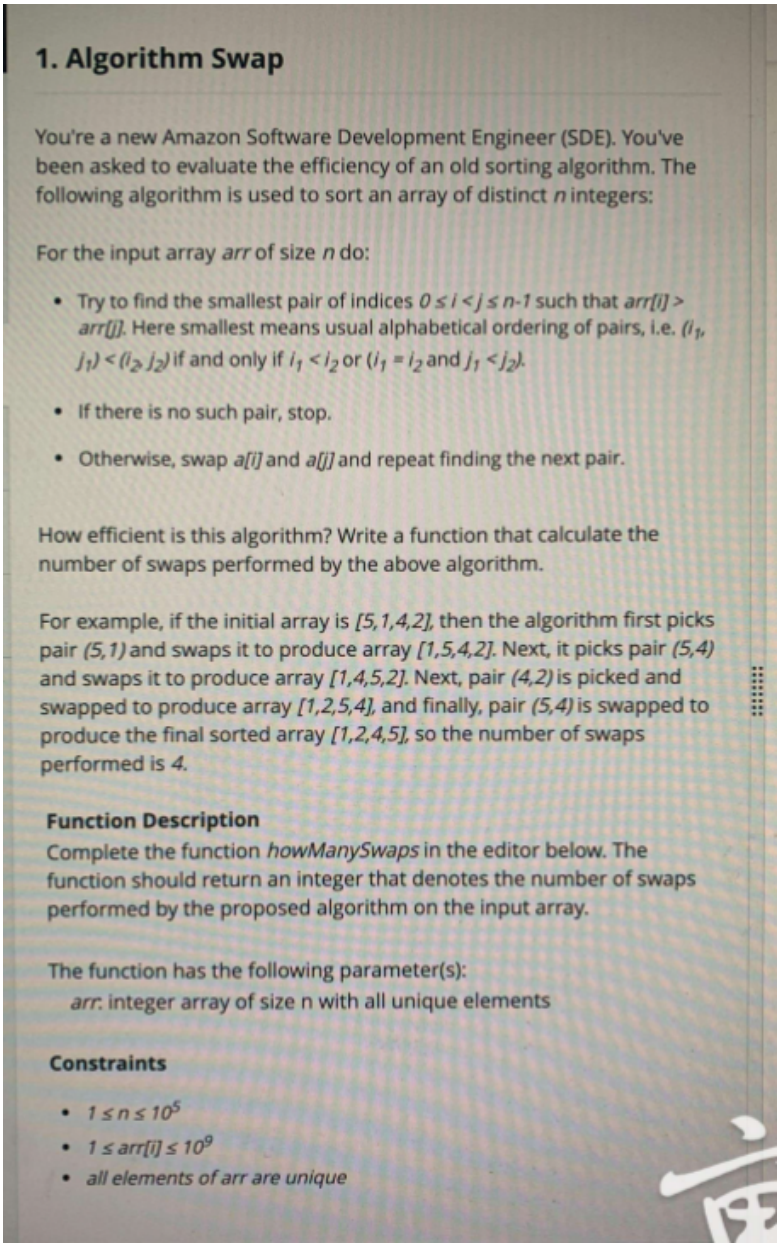
Returns:

int: an integer that denotes the sum of the values calculated

Constraints

- $2 \le n \le 10^5$
- $1 \le m \le 10^5$
- $1 \le p, q \le n$
- $p \ne q$

14. 1041. Robot Bounded in circle ++++++++++
15. Movies on Flight +++
16. Algorithm Swap /\315. Count of Smaller Numbers After Self



1. Algorithm Swap

You're a new Amazon Software Development Engineer (SDE). You've been asked to evaluate the efficiency of an old sorting algorithm. The following algorithm is used to sort an array of distinct *n* integers:

For the input array *arr* of size *n* do:

- Try to find the smallest pair of indices $0 \le i < j \le n-1$ such that *arr[i] > arr[j]*. Here smallest means usual alphabetical ordering of pairs, i.e. $(i_1, j_1) < (i_2, j_2)$ if and only if $i_1 < i_2$ or $(i_1 = i_2$ and $j_1 < j_2)$.
- If there is no such pair, stop.
- Otherwise, swap *a[i]* and *a[j]* and repeat finding the next pair.

How efficient is this algorithm? Write a function that calculate the number of swaps performed by the above algorithm.

For example, if the initial array is *[5,1,4,2]*, then the algorithm first picks pair *(5,1)* and swaps it to produce array *[1,5,4,2]*. Next, it picks pair *(5,4)* and swaps it to produce array *[1,4,5,2]*. Next, pair *(4,2)* is picked and swapped to produce array *[1,2,5,4]*, and finally, pair *(5,4)* is swapped to produce the final sorted array *[1,2,4,5]*, so the number of swaps performed is *4*.

Function Description

Complete the function *howManySwaps* in the editor below. The function should return an integer that denotes the number of swaps performed by the proposed algorithm on the input array.

The function has the following parameter(s):

arr: integer array of size n with all unique elements

Constraints

- $1 \le n \le 10^5$
- $1 \le arr[i] \le 10^9$
- *all elements of arr are unique*

    - https://leetcode.com/discuss/interview-question/1031247/Amazon-or-OA-or-No.-of-swaps

17. 1197. Minimum Knight Moves ++
18. 937. Reorder Data in Log Files ++
19. Demolition Robot+++

https://leetcode.com/discuss/interview-question/1033264/amazon-oa-1-year-experienced-for-sde1

3. Demolition Robot

You are in charge of preparing a recently purchased lot for one of Amazon's new building. The lot is covered with trenches and has a single obstacle that needs to be taken down before the foundation can be prepared for the building. The demolition robot must remove the obstacle before progress can be made on the building.

Write an algorithm to determine the minimum distance required for the demolition robot to remove the obstacle.

Assumptions:
The lot is flat, except for trenches, and can be represented as a two-dimensional grid.
The demolition robot must start from the top-left corner of the lot, which is always flat, and can move one block up, down, left, or right at a time.
The demolition robot cannot enter trenches and cannot leave the lot.
The flat areas are represented as 1, areas with trenches are represented by 0 and the obstacle is represented by 9.

Input
The input to the function/method consists of one argument:
lot, representing the two-dimensional grid of integers.

Output
Return an integer representing the minimum distance traversed to remove the obstacle else return -1.

Constraints
$1 \leq rows, columns \leq 10^3$

Example

Input:
lot =
[[1, 0, 0],
[1, 0, 0],
[1, 9, 1]]

Output:
3

Explanation:
Starting from the top-left corner, the demolition robot traversed the cells (0,0) -> (1,0) -> (2,0) -> (2,1). The robot traversed the total distance 3 to remove the obstacle.

So, the output is 3.

20. 973. K Closest Points to Origin +++ /Amazon Fresh Deliveries MaxHeap
21. 1099. Two Sum Less Than K
22. TransactionLog ++++++

- https://leetcode.com/discuss/interview-question/989768/Amazon-or-OA-2020-or-Transaction-logs
- https://leetcode.com/discuss/interview-question/912928/amazon-oa-2020-gifting-groups-friend-circle-lc-d

oesnt-solve-this

23. Optimize Box weight +++
24. Storage Optimzation +++

3. Storage Optimization

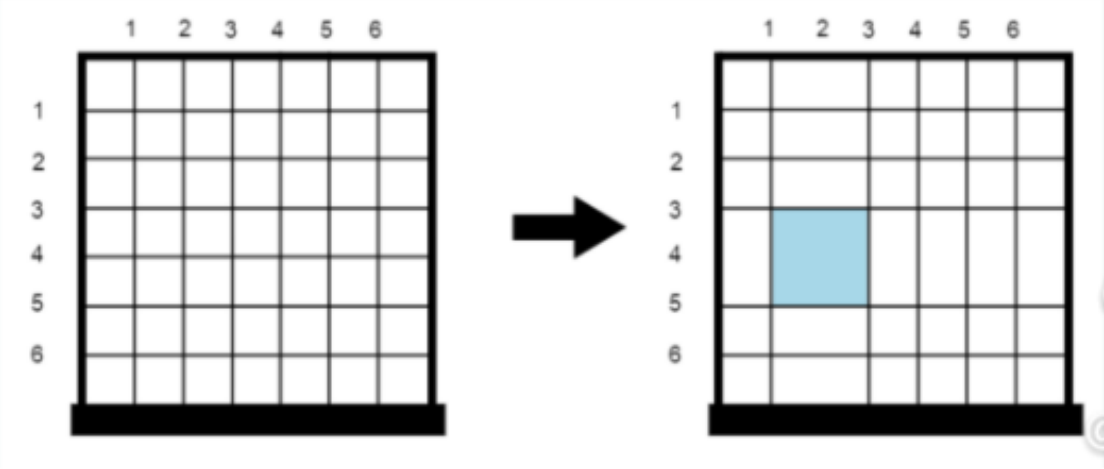
< 上一张

Amazon is experimenting with a flexible storage system for their warehouses. The storage unit consists of a shelving system which is one meter deep with removable vertical and horizontal separators. When all separators are installed, each storage space is one cubic meter (1' x 1' x 1'). Determine the volume of the largest space when a series of horizontal and vertical separators are removed.

Example

n = 6
m = 6
h = [4]
v = [2]

Consider the diagram below. The left image depicts the initial storage unit with *n = 6* horizontal and *m = 6* vertical separators, where the volume of the largest storage space is *1 × 1 x 1*. The right image depicts that unit after the fourth horizontal and *second vertical separators are removed. The maximum storage volume for that unit is then 2 × 2 x 1 = 4 cubic meters:*



Function Description

上一张 plete the function storage in the editor below.

storage has the following parameter(s):

int n: integer, the number of horizontal separators initially
int m: integer, the number of vertical separators initially
int h[x]: an array of integers, the horizontal separators to remove
int v[y]: an array of integers, the vertical separators to remove

Returns:

int: a long integer denoting the volume of the largest item that can be stored in the unit.

Constraints

- $1 \le n, m \le 10^5$
- $0 < x \le n$
- $0 < y \le m$
- $1 \le h[i] \le n$, where $1 \le i \le n$.
- $1 \le v[j] \le m$, where $1 \le j \le m$.
- The values in array *h* are distinct.
- The values in array *v* are distinct.

25. Shopping Option/Jeans,Shoes,Skirts, and Tops +++

- 每个list选一个，最后加起来价格小于等于budget的方案有几个
  list1 = [4]
  list2 = [3, 4]
  list3 = [1, 2]
  list4 = [2, 1]
  budget = 10
  最后 return 4， 因为有四种 组合， [4,3,1,1], [4,3,1,2], [4,3,2,1], [4,4,1,1]

```
public static int waystoBuy(int[] jeans,int[] shoes,int[] skirts,int[] tops,int budget){
        List<Integer> sumJeansShoes = new ArrayList<>();
        for(int i=0;i<jeans.length;i++){
            for(int j=0;j<shoes.length;j++){
                if(jeans[i]+shoes[j]<budget)
                    sumJeansShoes.add(jeans[i]+shoes[j]);
            }
        }
        Collections.sort(sumJeansShoes);
        List<Integer> sumSkirtsTops = new ArrayList<>();
```

```
        for(int i=0;i<skirts.length;i++){
            for(int j=0;j<tops.length;j++){
                if(skirts[i]+tops[j]<budget)
                    sumSkirtsTops.add(skirts[i]+tops[j]);
            }
        }
        Collections.sort(sumSkirtsTops);
        int numways=0;
        for(int i=0;i<sumJeansShoes.size();i++){
            for(int j=0;j<sumSkirtsTops.size();j++){
                if(sumJeansShoes.get(i) + sumSkirtsTops.get(j)<=budget)
                    numways++;
                else
                    break;
            }
        }

            return numways;
        }
```

26. 239 Sliding Window Maximum +++
27. other

- https://www.1point3acres.com/bbs/thread-546575-1-1.html 2019-8
- number game / 1799. Maximize Score After N Operations -- Hard
- 1335. Minimum Difficulty of a Job Schedule -- Hard / Minimum Container Size
- 1761. Minimum Degree of a Connected Trio in a Graph / Shopping Pattern