

Adobe Interview Problems

1) Given two integers n and k , return all possible combinations of k numbers out of $1\ 2\ 3\ \dots n$.

Make sure the combinations are sorted.

To elaborate,

Within every entry, elements should be sorted. $[1, 4]$ is a valid entry while $[4, 1]$ is not.

Entries should be sorted within themselves.

Example:

If $n = 4$ and $k = 2$, a solution is:

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

2) Given n non-negative integers a_1, a_2, \dots, a_n ,

where each represents a point at coordinate (i, a_i) .

' n ' vertical lines are drawn such that the two endpoints of line i is at (i, a_i) and $(i, 0)$.

Find two lines, which together with x -axis forms a container, such that the container contains the most water.

Your program should return an integer which corresponds to the maximum area of water that can be contained (Yes, we know maximum area instead of maximum volume sounds weird. But this is 2D plane we are working with for simplicity).

Note: You may not slant the container.

Example:

Input: $[1, 5, 4, 3]$

Output: 6

Explanation: 5 and 3 are distance 2 apart. So size of the base = 2. Height of container = $\min(5, 3) = 3$.

So total area = $3 * 2 = 6$

3) Give a $N \times N$ square matrix, return an array of its anti-diagonals. Look at the example for more details.

Example:

Input 1:

1 2 3

4 5 6

7 8 9

Output 1:

```
[  
  [1],  
  [2, 4],  
  [3, 5, 7],  
  [6, 8],  
  [9]  
]
```

Input 2:

1 2

3 4

Output 2:

```
[  
  [1],  
  [2, 3],  
  [4]  
]
```

4) Write a function that takes an unsigned integer and returns the number of 1 bits it has.

Example:

The 32-bit integer 11 has binary representation 00000000000000000000000000001011 so the function should return 3.

5) Design and implement a data structure for LRU (Least Recently Used) cache. It should support the following operations: get and set.

get(key) - Get the value (will always be positive) of the key if the key exists in the cache, otherwise return -1.

set(key, value) - Set or insert the value if the key is not already present. When the cache reaches its capacity, it should invalidate the least recently used item before inserting the new item.

The LRU Cache will be initialized with an integer corresponding to its capacity. Capacity indicates the maximum number of unique keys it can hold at a time.

Definition of “least recently used”: An access to an item is defined as a get or a set operation of the item. “Least recently used” item is the one with the oldest access time.

NOTE: If you are using any global variables, make sure to clear them in the constructor.

Example:

Input:

```
capacity = 2
set(1, 10)
set(5, 12)
get(5)    returns 12
get(1)    returns 10
get(10)   returns -1
set(6, 14) this pushes out key = 5 as LRU is full.
get(5)    returns -1
```