Som

29 June 2021 14:17

One problem (dictonary in python)

First technique which we were using to find out the indices of the target value of array was inefficient
It has O(n^2) runtime and O(1) space complexity
To improve that we use the hash tables (dictionary) in python
We store the required number and their indices in dictionary
After storing that we do check while every time storing the required number in the dictionary is the value already available there if it's there then we are done and we find the indices since we've stored keys as the required value and the indices as the values to that
we return the indices of the required value if it's available there and the value which we were going to find the pair of indices there

Code for same

```
def effi(arr,target):
key_values = {}
for val in range(len(arr)):
if arr[val] in key_values.keys():
print(key_values[arr[val]],val)
return key_values[arr[val]],val
else:
key_values[target - arr[val]] = val
```

Two pointers in python

Another problem where we wanted to find the maximum area of in between the 2 lines where the distance is being seperated by 1 unit and every of wall is given in form of array

We can solve this problem by the traditional way having 2 for loops each iterate through each other find the every other possible case and store the maximum area value
But this problem is again going to give us the time complexity of O(n^2)
So for finding the solution of this problem we have to set the two pointers one at the start and one at the end
Since the wall is gonna choosen to be minimum one for making the rectangle
For this purpose we were trying to change the minimum no as max as possible

The formal solution is this

Code for same

```
def area_of_container_effective(a):
first = 0
last = len(a) - 1
maximum =0
while first < last:
area = min(a[first],a[last]) * (last -first)
maximum = max(maximum , area)
if a[first] > a[last] :
last -= 1
elif a[first] < a[last]:
first += 1
```

```
    return maximum
```

Here we are using kind of binary search and at the end of the graphs we can't have the solution if the graph is having only 1 or none elements then also we don't have any solution