Leet code

Feb 6 2023

Q1. Given the array nums consisting of 2n elements in the form [x1,x2,...,xn,y1,y2,...,yn]. Return the array in the form [x1,y1,x2,y2,...,xn,yn].

In [8]:
```
class Solution:
    def shuffle(self, nums: list[int], n: int) -> list[int]:
        l1 = nums[0:n]
        l2 = nums[n:]
        final = []
        for i in range(n):
            final.append(l1[i])
            final.append(l2[i])
        return final

S = Solution();
S.shuffle([2,5,1,3,4,7], 3)
```

Out[8]:
```
[2, 3, 5, 4, 1, 7]
```

Q2. Given two strings s and p, return an array of all the start indices of p's anagrams in s. You may return the answer in any order.

An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.{Feb 5, 2023}

In [4]:
```
class Solution:
    def findAnagrams(self, s: str, p: str) -> list[int]:
        if len(p)>len(s):
            return([])
        i = 0
        j = len(p)
        l = []
        while j<= len(s):
            if sorted(p) == sorted(s[i:j]):
                l.append(i)
                i = i + 1
                j = j + 1
            else:
                i = i + 1
                j = j + 1
        print(l)
        return l
S = Solution()
S.findAnagrams("eidbaooo","ab")
```

Out[4]:
```
[3]
[3]
```

Q2. Given two strings s1 and s2, return true if s2 contains a permutation of s1, or false otherwise. In other words, return true if one of s1's permutations is the substring of s2. [Feb 4 2023]

In [11]:
```
class Solution:
    def checkInclusion(self, p: str, s: str) -> bool:
```

```
        if len(p)>len(s):
            return False
        i = 0
        j = len(p)
        setbit = 0
        while j<= len(s):
            if sorted(p) == sorted(s[i:j]):
                i = i + 1
                j = j + 1
                setbit = 1
                if setbit == 1:
                    return True
            else:
                i = i + 1
                j = j + 1

S = Solution()
S.checkInclusion("ab","eidbaooo")
```

Out[11]: True