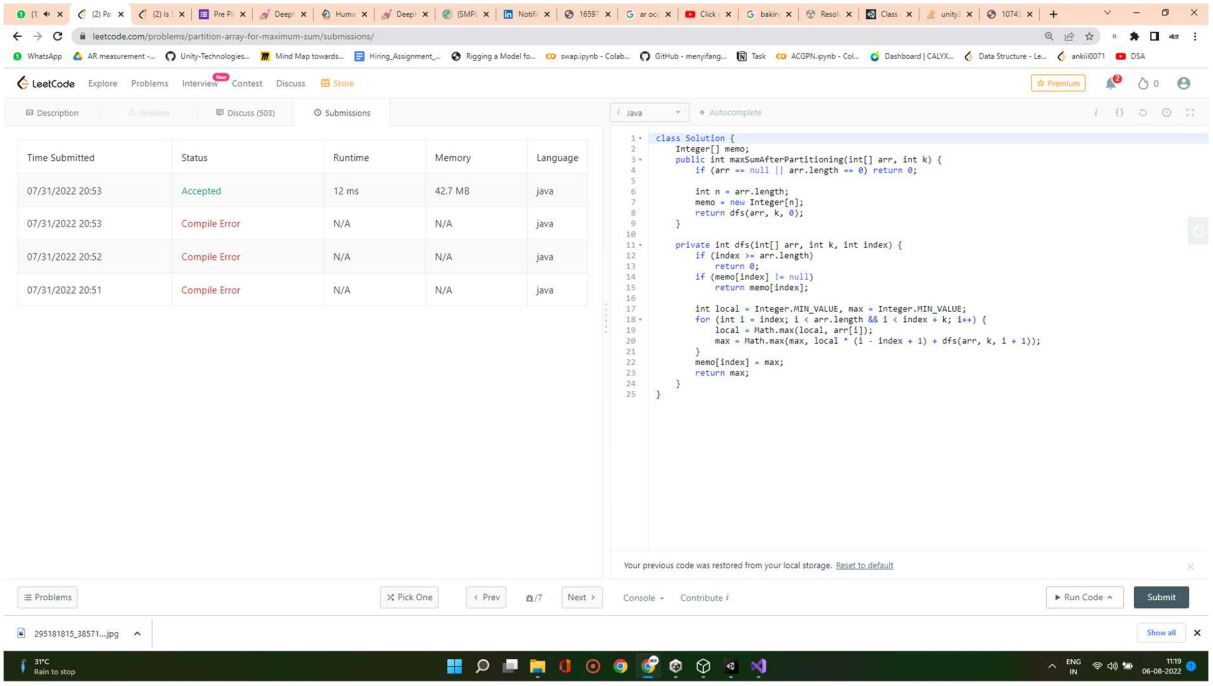
Leetcode Solution



leetcode.com/problems/partition-array-for-maximum-sum/submissions/

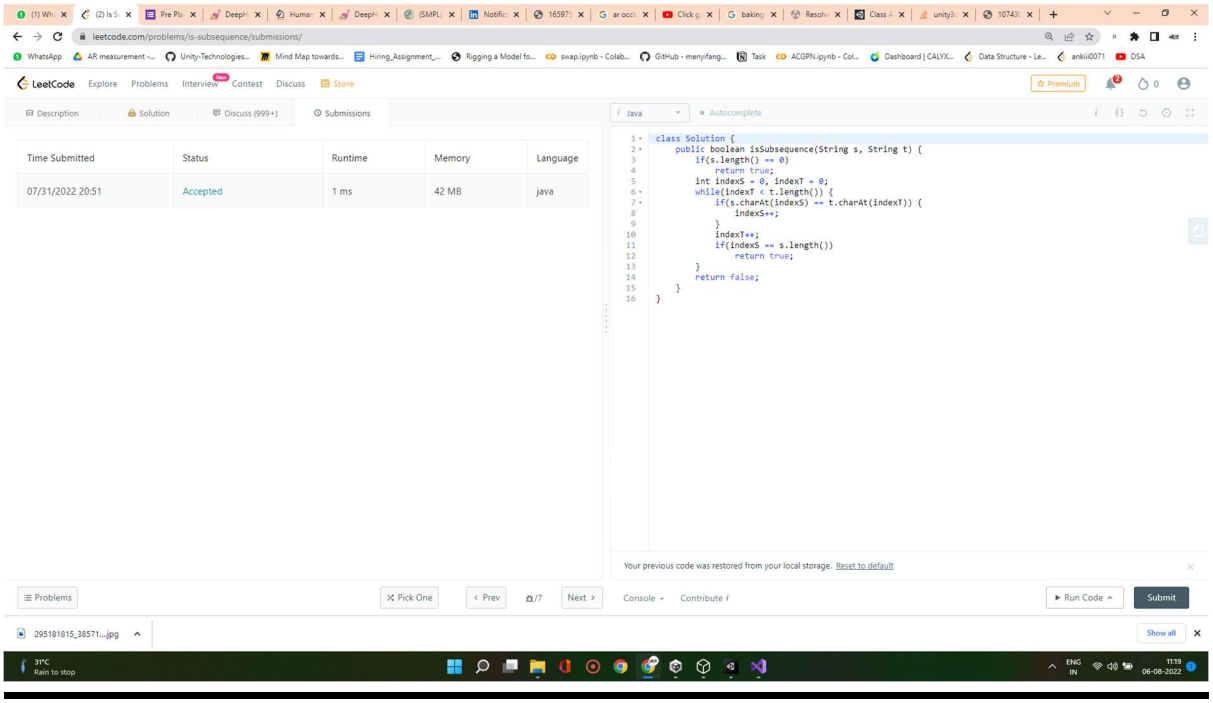
Time Submitted	Status	Runtime	Memory	Language
07/31/2022 20:53	Accepted	12 ms	42.7 MB	java
07/31/2022 20:53	Compile Error	N/A	N/A	java
07/31/2022 20:52	Compile Error	N/A	N/A	java
07/31/2022 20:51	Compile Error	N/A	N/A	java

```java
class Solution {
    Integer[] memo;
    public int maxSumAfterPartitioning(int[] arr, int k) {
        if (arr == null || arr.length == 0) return 0;

        int n = arr.length;
        memo = new Integer[n];
        return dfs(arr, k, 0);
    }

    private int dfs(int[] arr, int k, int index) {
        if (index >= arr.length)
            return 0;
        if (memo[index] != null)
            return memo[index];

        int local = Integer.MIN_VALUE, max = Integer.MIN_VALUE;
        for (int i = index; i < arr.length && i < index + k; i++) {
            local = Math.max(local, arr[i]);
            max = Math.max(max, local * (i - index + 1) + dfs(arr, k, i + 1));
        }
        memo[index] = max;
        return max;
    }
}
```



leetcode.com/problems/is-subsequence/submissions/

Time Submitted	Status	Runtime	Memory	Language
07/31/2022 20:51	Accepted	1 ms	42 MB	java

```java
class Solution {
    public boolean isSubsequence(String s, String t) {
        if(s.length() == 0)
            return true;
        int indexS = 0, indexT = 0;
        while(indexT < t.length()) {
            if(s.charAt(indexS) == t.charAt(indexT)) {
                indexS++;
            }
            indexT++;
            if(indexS == s.length())
                return true;
        }
        return false;
    }
}
```