**DP codes**

1. **Subset sum code: Tabulation**

def isSubsetSum (self, N, arr, sum):

# code here

# subset=[N+1][sum+1]

# subset=[[0 for i in range(sum+1)] for j in range(N+1)]

subset=[][]

for i in range(N+1):

subset[i][0]=True

for j in range(1,sum+1):

subset[0][i]=False

for i in range(1,N+1):

for j in range(1,sum+1):

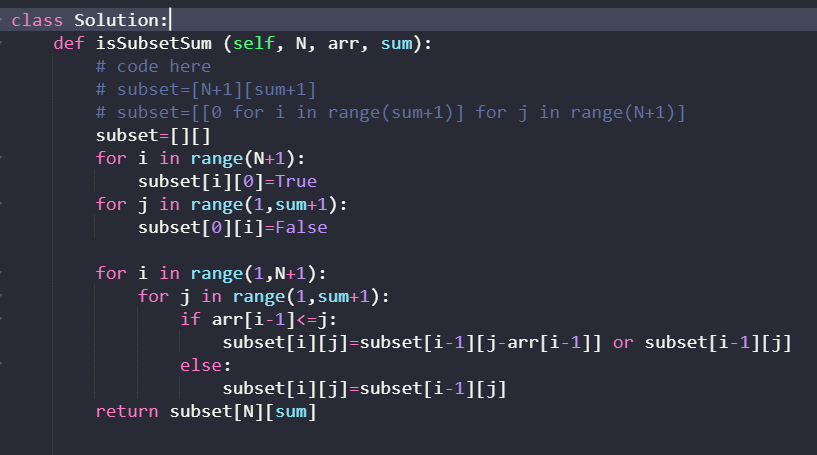
if arr[i-1]<=j:

subset[i][j]=subset[i-1][j-arr[i-1]] or subset[i-1][j]

else:

subset[i][j]=subset[i-1][j]

return subset[N][sum]



**b) Equal sum partition:**

def equalPartition(self, N, arr):

# code here

s=0

for i in arr:

s=s+i

if s%2!=0:

return 0

else:

subset=[[0 for i in range(s//2+1)] for j in range(N+1)]

for i in range(N+1):

subset[i][0]=True

for j in range(1,s//2+1):

subset[0][i]=False

for i in range(1,N+1):

for j in range(1,s//2+1):

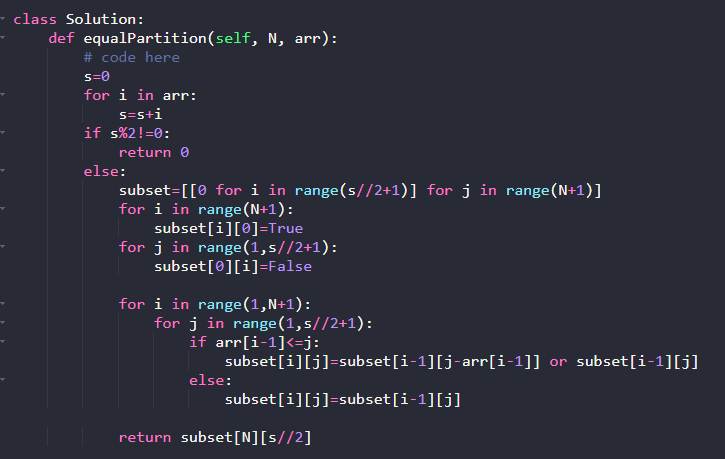
if arr[i-1]<=j:

subset[i][j]=subset[i-1][j-arr[i-1]] or subset[i-1][j]

else:

subset[i][j]=subset[i-1][j]

return subset[N][s//2]



**c) Count of Subset sum with given sum:**

def perfsum(arr,n,sum):  
 tab = [[0] \* (sum + 1) for i in range(n + 1)]  
 tab[0][0] = 1  
 for i in range(1, sum + 1):  
 tab[0][i] = 0  
  
 for i in range(1, n+1):  
 for j in range(sum + 1):  
 if arr[i-1] <= j:  
 tab[i][j] = tab[i-1][j] + tab[i-1][j-arr[i-1]]  
 else:  
 tab[i][j] = tab[i-1][j]  
 return tab[n][sum]  
  
if \_\_name\_\_=='\_\_main\_\_':  
 n,sum=input().split()  
 n,sum=int(n),int(sum)  
 arr=[int(x) for x in input().split()]  
 op=perfsum(arr,n,sum)  
 print(op)

sfsz

**d) Minimum subset sum difference:**

def minDifference(self, arr, n):  
# # code here  
 sum=0  
 for i in arr:  
 sum=sum+i  
 t=[[0 for i in range(sum+1)] for i in range(n+1)]  
 for i in range(n+1):  
 t[i][0]=1  
 for i in range(1,sum+1):  
 t[0][i]=0  
 for i in range(1,n+1):  
 for j in range(1,sum+1):  
 if(arr[i-1]<=j):  
 t[i][j]=t[i-1][j-arr[i-1]] or t[i-1][j]  
 else:  
 t[i][j]=t[i-1][j]  
 mm=sum  
 for i in range((sum//2)+1):  
 if t[n][i]:  
 mm=min(mm,sum-(2\*i))  
 return mm

or

for i in range((sum // 2), -1, -1):  
 if t[n][i]:

return (sum - (2 \* i))

**e) Count of subsets with given difference:**

def diff(arr,d):  
 sm=sum(arr)  
 n=len(arr)  
 rang=(d+sm)//2  
 t=[[0]\*(rang+1) for j in range(n+1)]  
 for i in range(n+1):  
 t[i][0]=1  
 for i in range(1,n+1):  
 for j in range(1,rang+1):  
 if arr[i-1]<=j:  
 t[i][j]=t[i-1][j-arr[i-1]] + t[i-1][j]  
 else:  
 t[i][j]=t[i-1][j]  
  
 return t[n][rang]  
  
print(diff([1,1,2,3,4,5],2))

sz

**Strings:**

**Longest Common Subsequence:**

def lcs(s1,s2,n,m):

if(n==0 or m==0):

return 0

elif(s1[n-1]==s2[m-1]):

return 1+lcs(s1,s2,n-1,m-1)

else:

return max(lcs(s1,s2,n-1,m),lcs(s1,s2,n,m-1))

s1=input()

s2=input()

print(lcs(s1,s2,len(s1),len(s2)))

