

# Knapsack

```
def knapsack_possible(target_length, bar_lengths):
    p = len(bar_lengths)
    DP = [[False] * (target_length + 1) for _ in range(p + 1)]

    for i in range(p + 1):
        DP[i][0] = True

    for i in range(1, p + 1):
        for j in range(1, target_length + 1):
            if bar_lengths[i - 1] > j:
                DP[i][j] = DP[i - 1][j]
            else:
                DP[i][j] = DP[i - 1][j] or DP[i - 1][j - bar_lengths[i - 1]]


    return DP[p][target_length]

def main():
    t = int(input())

    for case_num in range(1, t + 1):
        target_length = int(input())
        p = int(input())
        bar_lengths = list(map(int, input().split()))

        if knapsack_possible(target_length, bar_lengths):
            print("YES")
        else:
            print("NO")

if __name__ == "__main__":
    main()
```

Status	Time	Length	Lang	Submitted	Open	Share text 	RemoteRunId
Accepted	20ms	832	PYTH3 3.5.1	2024-10-19 15:59:45	<input checked="" type="checkbox"/>	<input type="checkbox"/>	29895521

```
1 def knapsack_possible(target_length, bar_lengths):
2     p = len(bar_lengths)
3     DP = [[False] * (target_length + 1) for _ in range(p + 1)]
4
5     for i in range(p + 1):
6         DP[i][0] = True
7
8     for i in range(1, p + 1):
9         for j in range(1, target_length + 1):
10             if bar_lengths[i - 1] > j:
11                 DP[i][j] = DP[i - 1][j]
12             else:
13                 DP[i][j] = DP[i - 1][j] or DP[i - 1][j - bar_lengths[i - 1]]
14
15     return DP[p][target_length]
16
17 def main():
18     t = int(input())
19
20     for case_num in range(1, t + 1):
21         target_length = int(input())
22         p = int(input())
23         bar_lengths = list(map(int, input().split()))
24
25         if knapsack_possible(target_length, bar_lengths):
26             print("YES")
27         else:
28             print("NO")
29
30 if __name__ == "__main__":
31     main()
```

[Copy](#) [Python](#)