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	☑	☐	29895521

```python
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()
```