187. Consider a set of keys 10,12,16,21 with frequencies 4,2,6,3 and the respective probabilities. Write a Program to construct an OBST in a programming language of your choice. Execute your code and display the resulting OBST, its cost and root matrix.

Input N =4, Keys = {10,12,16,21} Frequencies = {4,2,6,3}

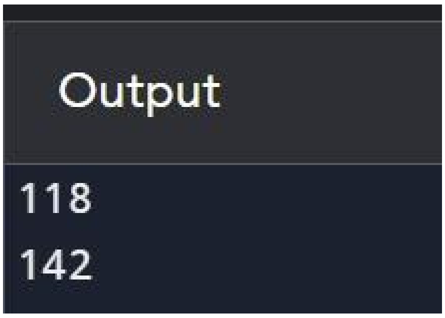
Output : 26

Program:

```
def optimal_bst(keys, freq):
    n = len(keys)
    cost = [[0 for _ in range(n)] for _ in range(n)]
    root = [[0 for _ in range(n)] for _ in range(n)]

    for i in range(n):
        cost[i][i] = freq[i]
        root[i][i] = i

    for L in range(2, n + 1):
        for i in range(n - L + 1):
            j = i + L - 1
            cost[i][j] = float('inf')

            for r in range(i, j + 1):
                c = cost[i][r - 1] if r > i else 0
                c += cost[r + 1][j] if r < j else 0
                c += sum(freq[i:j + 1])

                if c < cost[i][j]:
                    cost[i][j] = c
                    root[i][j] = r

    return cost[0][n - 1], root

# Test cases
keys1 = [10, 12]
```

```
freq1 = [34, 50]

keys2 = [10, 12, 20]

freq2 = [34, 8, 50]


result1, _ = optimal_bst(keys1, freq1)

result2, _ = optimal_bst(keys2, freq2)


print(result1)  # Output: 118

print(result2)  # Output: 142
```

Output:



```
118
142
```

Timecomplexity:O(V+E)