

LAB-10

THROWING N DICE

CODE:

```
def numberofpossibilities(n):
```

```
    return 6 ** n
```

```
n = 3
```

```
print(f"The number of possibilities when throwing {n} dice is ",numberofpossibilities(n))
```

OUTPUT:

```
>>> = RESTART: C:/Users/bored/AppData/Local/Programs/Python/Python312/throwing n dice.py
>>> The number of possibilities when throwing 3 dice is  216
```

TSP USING DP

CODE:

```
n = 4
```

```
dist = [[0, 0, 0, 0, 0], [0, 0, 10, 15, 20], [
        0, 10, 0, 25, 25], [0, 15, 25, 0, 30], [0, 20, 25, 30, 0]]
```

```
memo = [[-1]*(1 << (n+1)) for _ in range(n+1)]
```

```
def fun(i, mask):
```

```
    if mask == ((1 << i) | 3):
```

```
        return dist[1][i]
```

```
    if memo[i][mask] != -1:
```

```
        return memo[i][mask]
```

```
    res = 10**9
```

```
    for j in range(1, n+1):
```

```
        if (mask & (1 << j)) != 0 and j != i and j != 1:
```

```
            res = min(res, fun(j, mask & ~(1 << i)) + dist[j][i])
```

```
    memo[i][mask] = res
```

```
    return res
```

```
ans = 10**9
```

```
for i in range(1, n+1):
```

```
    ans = min(ans, fun(i, (1 << (n+1))-1) + dist[i][1])
```

```
print("The cost of most efficient tour = " + str(ans))
```

OUTPUT:

```
>>> = RESTART: C:/Users/bored/AppData/Local/Programs/Python/Python312/tsp using dp.py
>>> The cost of most efficient tour = 80
>>> |
```

OBST USING DP

CODE:

```
def optCost(freq, i, j):
    if j < i:
        return 0

    if j == i:
        return freq[i]

    fsum = Sum(freq, i, j)

    Min = 999999999999

    for r in range(i, j + 1):
        cost = (optCost(freq, i, r - 1) +
                optCost(freq, r + 1, j))

        if cost < Min:
            Min = cost

    return Min + fsum

def optimalSearchTree(keys, freq, n):
    return optCost(freq, 0, n - 1)

def Sum(freq, i, j):
    s = 0

    for k in range(i, j + 1):
        s += freq[k]

    return s

if __name__ == '__main__':
    keys = [10, 12, 20]
    freq = [34, 8, 50]
    n = len(keys)
```

```
print("Cost of Optimal BST is",  
      optimalSearchTree(keys, freq, n))
```

OUTPUT:

```
>>> = RESTART: C:/Users/bored/AppData/Local/Programs/Python/Python312/obst using dp.  
py  
Cost of Optimal BST is 142  
>>> |
```