

## Coin Change & 0/1 knapsack DP

মনে করো তোমার কাছে  $n$  টা ভিন্ন ভিন্ন কয়েন আছে, কয়েনগুলোর ভ্যালুকে  $C_0, C_1 \dots C_{n-1}$  দিয়ে প্রকাশ করা যায়। আর তোমাকে একটা অ্যামাউন্ট দেয়া আছে  $W$ । এখন তোমাকে বলতে সর্বনিম্ন কয়টা কয়েন ব্যবহার করে তুমি  $W$  অ্যামাউন্টটা বানাতে পারবে। প্রতিটা ভ্যালুর কয়েন আছে মাত্র ১টা করে।

একটা উদাহরণ দেখি। ধরা যাক কয়েনগুলোর ভ্যালু হলো  $C = \{2, 5, 9, 13, 15\}$  টাকা। এখন তুমি এই কয়েনগুলো দিয়ে  $W = 22$  বানাতে চাইলে একটা উপায় হলো  $15 + 5 + 2$ , এক্ষেত্রে কয়েন লাগছে ৩টা। কিন্তু তুমি চাইলে ২টা কয়েন ব্যবহার করেও 22 বানাতে পারো ( $9 + 13$ )। আমাদের টার্গেট কয়েন ব্যবহার মিনিমাইজ করা।

- $i$  তম কয়েনটাকে ব্যবহার করলে আমাদের টার্গেট বাকি থাকে  $W - C_i$ । তাহলে পরবর্তি সাবপ্রবলেম হবে  $f(i + 1, W - C_i)$ ।
- $i$  তম কয়েনটাকে ব্যবহার না করলে আমাদের টার্গেট পরিবর্তন হবে না। তাহলে পরবর্তি সাবপ্রবলেম হবে  $f(i + 1, W)$ ।

```
int C[MAX_N];
int mem[MAX_N][MAX_W];
int n;

int f(int i, int W) {
    if (W < 0) return INF;
    if (i == n) {
        if (W == 0) return 0;
        return INF;
    }

    if (mem[i][W] != EMPTY_VALUE) {
        return mem[i][W];
    }

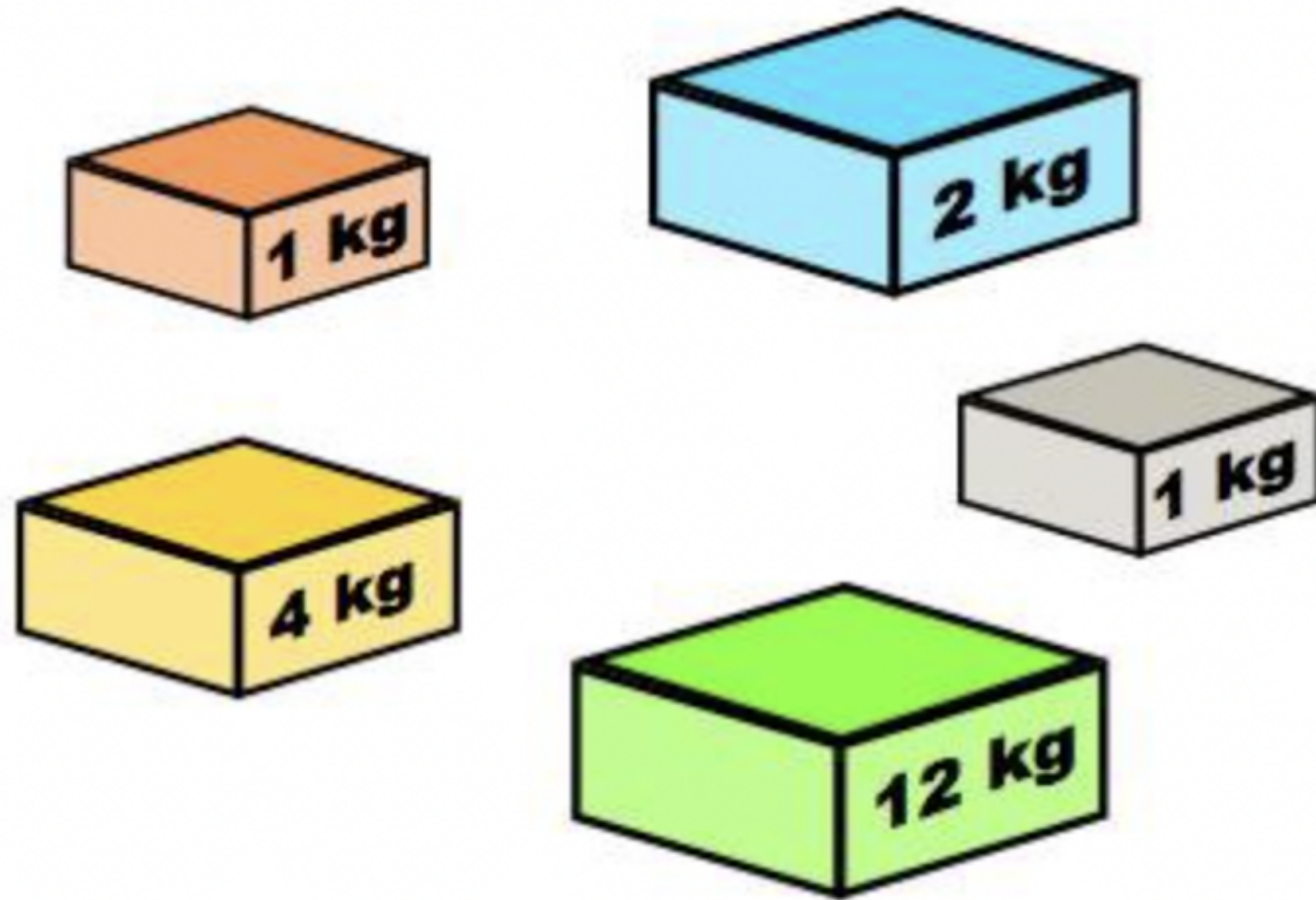
    int res_1 = 1 + f(i + 1, W - C[i]);
    int res_2 = f(i + 1, W);

    mem[i][W] = min(res_1, res_2);

    return mem[i][W];
}
```



# How to solve the Knapsack Problem with dynamic programming



“given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.”

এখানে ইনপুট হিসাবে দেয়া হবে দুটি অ্যারে  $P$  এবং  $W$ ।  $i$  তম বস্তুর দাম  $P_i$  এবং ওজন  $W_i$ ।

আমাদের সাবপ্রবলেম আগের মতোই হবে  $f(i, C)$  যা দিয়ে বুঝাবে  $i$  থেকে শুরু করে  $n - 1$  তম আইটেম গুলো দিয়ে পাওয়া সর্বোচ্চ প্রফিট।  
সেখান থেকে আমাদের দুইটা চয়েস

- $i$  তম আইটেম ব্যাগে না ভরা, তাহলে পরবর্তী সাবপ্রবলেম হবে  $f(i + 1, C)$
- $i$  তম আইটেম নেয়া, পরবর্তী সাবপ্রবলেম হবে  $f(i + 1, C - W_i)$ ।



- $i$  তম আইটেম ব্যাগে না ভরা, তাহলে পরবর্তী সাবপ্রবলেম হবে  $f(i + 1, C)$
- $i$  তম আইটেম নেয়া, পরবর্তী সাবপ্রবলেম হবে  $f(i + 1, C - W_i)$ ।

আমাদেরকে এই দুইটার মধ্যে থেকে বড়টাকে নিতে হবে। ২য় ক্ষেত্রে প্রফিট হবে  $P_i$ , সেটাও যোগ করতে হবে।

$$f(i, C) = \max(f(i + 1, C), f(i + 1, C - W_i) + P_i)$$

















