

Dynamic Planning

Knapsack

[例題](#) | [参考](#)

n 個の品物があり、 i 番目の品物のそれぞれ重さと価値が $w[i]$ 、 $v[i]$ となっている ($i = 0, 1, \dots, n-1$)。これらの品物から重さの総和が W を超えないように選んだときの、価値の総和の最大値を求めよ。

方針

```
dp[i][j]  >>>  i 番目までの品物の中で重さの総和が j 以下となるように選んだときの最大価値
```

漸化式

```
if j >= w[i] then
    dp[i+1][j] = max{ dp[i][j - w[i]] + v[i], dp[i][j] }
else
    dp[i+1][j] = dp[i][j]
```

初期条件

```
for w in W
    dp[0][w] = 0
```

Number Partitioning 1

[例題](#) | [参考](#)

n 個の正の整数 $a[0], a[1], \dots, a[n-1]$ と正の整数 A が与えられる。これらの整数から何個かの整数を選んで総和が A になるようにすることが可能か判定せよ。

方針

```
dp[i][j]  >>>  i番目までの整数の中からいくつか選んで総和をjとすることが可能かどうか
```

漸化式

```
if j >= a[i] then
    dp[i+1][j] = dp[i][j-a[i]] | dp[i][j]
else
    dp[i+1][j] = dp[i][j]
```

初期条件

```
for j in A
    if j = 0 then
        dp[0][j] = True
    else
        dp[0][j] = False
```