# Week 5 - B

Roy

# Week5 - B

- Uva 1213 - Sum of Different Primes
- DP 題

# 題目敘述

- 給兩個正整數N<=1120跟K<=14，問用K個質數，有幾種質數和=N
  - e.g. 5 = 2 + 3

# 想法

- 背包問題改
- dp[i][j]
  - i 是「目前總和」
  - j 是「幾個質數組成」

# 轉移式

- dp[i][j] = dp[i][j] + dp[i-p[k]][j-1]
  - p[k] 第 k 個質數
  - 當前的種數(i, j) 加上 (i-p[k], j-1)
    - 扣掉當前質數(p[k])、數量減一(j-1)的情況

  - 為什麼是用加的
    - 有多個質數 (2, 3, 5, ...) 可產生的種數

# 說明

- 為了說明方便，以下圖例以 n=5 k=2 作為範例

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=5 j=2

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | | | |
| 2 | | | |
| 3 | | 0 | |
| 4 | | | |
| 5 | | | 0 |

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | | | |
| 2 | | | |
| 3 | 0 | 0 | |
| 4 | | | |
| 5 | | 0 | 0 |

i=5 j=1

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=4 j=2

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | | | |
| 2 | | 0 | |
| 3 | 0 | 0 | |
| 4 | | | 0 |
| 5 | | 0 | 0 |

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=4 j=1

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | | | |
| 2 | 0 | 0 | |
| 3 | 0 | 0 | |
| 4 | | 0 | 0 |
| 5 | | 0 | 0 |

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | | 0 | |
| 2 | 0 | 0 | |
| 3 | 0 | 0 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 0 |

i=3 j=2

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=3 j=1

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | 0 | 0 | |
| 2 | 0 | 0 | |
| 3 | 0 | 0 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 0 |

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 0 |

i=2 j=2

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 0 |

i=2 j=1

```
dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 0 |

i=2 j=1

i >= pv[k] = 2
=> k++

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 1 |

i=5 j=2

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 1 |

i=5 j=1

省略步驟

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 1 |

i=3 j=1

```
dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 1 |

i=3 j=1

i >= pv[k] = 3
=> k++

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | | 0 | 0 |
| 5 | | 0 | 1 |

i=5 j=2

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | | 0 | 0 |
| 5 | | 1 | 1 |

i=5 j=1

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | | 0 | 0 |
| 5 | | 1 | 1 |

i=5 j=1

程式結束

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

# 方向？

- 為什麼 i 的方向是反的？
  - 如果正著做，會有出現重複質數的情況
    - e.g. 4 = 2 + 2

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = 5; i >= pv[k]; i--)
    for(int j = 2; j >= 1; j--)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

j 正/反不影響結果

# DP 正著做

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=2 j=1

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = pv[k]; i <= 5; i++)
    for(int j = 1; j <= 2; j++)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

# DP 正著做

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | | | |
| 2 | | 1 | |
| 3 | | | |
| 4 | | | |
| 5 | | | |

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=2 j=1

i-prime[k]=0

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = pv[k]; i <= 5; i++)
    for(int j = 1; j <= 2; j++)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

# DP 正著做

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | | | |
| 2 | | 1 | 0 |
| 3 | | | |
| 4 | | | |
| 5 | | | |

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=2 j=2

i-prime[k]=0

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = pv[k]; i <= 5; i++)
    for(int j = 1; j <= 2; j++)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

# DP 正著做

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | | 1 | 0 |
| 3 | | 0 | 0 |
| 4 | | | |
| 5 | | | |

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=3 j=2

i-prime[k]=1

省略步驟

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = pv[k]; i <= 5; i++)
    for(int j = 1; j <= 2; j++)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

# DP 正著做

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | | 0 | 0 |
| 4 | | 0 | 1 |
| 5 | | | |

| k | 0 | 1 | 2 |
|---|---|---|---|
| prime[k] | 2 | 3 | 5 |

i=4 j=2

i-prime[k]=1

省略步驟

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = pv[k]; i <= 5; i++)
    for(int j = 1; j <= 2; j++)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```

# DP 正著做

| dp[][] | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 2 | 0 | 1 | 0 |
| 3 | | 0 | 0 |
| 4 | | 0 | 1 |
| 5 | | | |

dp[4][2] = dp[4][2] + dp[2][1]

如果正著做
總和4，2個質數的情形
會是「總和2，1個質數」的情形 + 1

等於是 4 = 2 + 2
但本題不允許重複的質數和

```
dp[0][0] = 1;
for(int k = 0; k < pv.size(); k++)
  for(int i = pv[k]; i <= 5; i++)
    for(int j = 1; j <= 2; j++)
      dp[i][j] = dp[i][j] + dp[i-pv[k]][j-1];
```