

Зад. 1

Дадени са n различни вида монети - $\text{coins}[1..n]$. При условие, че имаме неограничен брой монети от всеки вид, да се намери броя различни начини да се получи дадена сума S .

Вход:

```
3
1 2 5
5
```

Изход:

```
4
//1+1+1+1+1
//1+1+1+2
//1+2+2
//5
```

```
d[r][c]←sum(d[r-1][c], d[r][c-coins[r-1]])
```

```
task1(coins[1..n], n, S):
```

```
  d[1..n+1][1..S+1] //init matrix
  for c←2 to S+1
    d[1][c]←0
  for r←1 to n+1
    d[r][1]←1
  for r←2 to n+1
    for c←2 to S+1
      d[r][c]←d[r-1][c]
      if c>coins[r-1] then
        d[r][c]←d[r][c]+d[r][c-coins[r-1]]
  return d[n+1][S+1]
```

Зад. 2

Дадени са n различни вида монети - $\text{coins}[1..n]$. Да се намери минималния брой монети, необходими да се получи дадена сума S .

- а) при условие, че имаме неограничен брой монети
- б) при условие, че имаме по 1 монета от вид

а)

Вход:

```
4
1 5 6 8
11
```

Изход:

```
2
```

```
//5+6
```

a)

```
d[r][c] ← min(d[r-1][c], 1+d[r][c-coins[r-1]])
```

task2a(coins[1..n], n, S):

```
d[1..n+1][1..S+1] ← [[∞, ..., ∞], ..., [∞, ..., ∞]]
```

```
for r ← 1 to n
```

```
    d[r][1] ← 0
```

```
for r ← 2 to n+1
```

```
    for c ← 2 to S+1
```

```
        d[r][c] ← d[r-1][c]
```

```
        if c > coins[r-1] and 1+d[r][c-coins[r-1]] < d[r][c] then
```

```
            d[r][c] ← 1+d[r][c-coins[r-1]]
```

```
return d[n+1][S+1]
```

б)

```
d[r][c] ← min(d[r-1][c], 1+d[r-1][c-coins[r-1]])
```

Формулата е 1:1 с разликата, че $d[r][c-coins[r-1]] \rightarrow d[r-1][c-coins[r-1]]$. семантиката е, че ни е гарантирано, че не сме използвали текущата монета.

Зад. 3

Даден е регулярен израз $regex[1..n]$ и дума $w[1..m]$. Да се провери дали думата се "чете" от регулярния израз. Нека също така за целта на задачата да разгледаме само подмножество на регулярните изрази и по-конкретно - тези, състоящи се от малки латински букви и мета символите '.' и '*'.

Примери:

regex: a.b

w: acb, aab, axb → TRUE

w: ab, acby, cb → FALSE

regex: a*b

w: b, ab, aab, aaab → TRUE

w: a, acb → FALSE

regex: a*b.*y

w: by, bly, ably, ablmy → TRUE

w: ay, ab → FALSE

$$d[i][j] \leftarrow \begin{cases} d[i-1][j-1], & w[i] = regex[j] \text{ or } regex[j] = '.' \\ d[i][j-2], & regex[j] = '*' \text{ and } d[i][j-2] = \text{TRUE} \\ d[i-1][j], & regex[j] = '*' \text{ and } (w[i] = regex[j-1] \text{ or } regex[j-1] = '.') \\ \text{FALSE,} & \text{else} \end{cases}$$

```

task3(regex[1..m], w[1..m], n, m):
  d[0..m][0..n]←[[TRUE, FALSE, ..., FALSE], ..., [FALSE, ..., FALSE]]
  for i←2 to n //with step 2
    d[0][i]←(d[0][i-2] and regex[i]='*')
  for i←1 to m
    for j←1 to n
      if regex[j]=w[i] or regex[j]='.' then
        d[i][j]←d[i-1][j-1]
      if j>2 and regex[j]='*' and d[i][j-2]=TRUE then
        d[i][j]←TRUE
      if regex[j]='*' and (w[i]=regex[j] or regex[j]='.') then
        d[i][j]←d[i-1][j]
  return d[m][n]

```

Зад. 4

Дадена е матрица $A[1..m][1..n] \in \{0, 1\}^{m \times n}$. Да се намери най-големия квадрат от единици и да се изведе от колко единици е съставен.

Пример:

```

0 1 0 1 1 0
1 1 1 1 1 1
1 0 0 1 1 1
1 1 0 1 1 1
1 1 1 1 1 1

```

Изход:

9

Остана за другия път..