

Design and Analysis of Algorithms

04-01

Dynamic Programming

Change Problem

Imran Ihsan

Assistant Professor, Department of Computer Science
Air University, Islamabad, Pakistan
www.imranihsan.com

Change Problem

1. Greedy Change
2. Recursive Change
3. Dynamic Programming

Change Problem

Find the minimum number of coins needed to make change



Change Problem – Formally

Input: An integer money and positive integers $\text{coin}_1, \dots, \text{coin}_d$.

Output: The minimum number of coins with denominations $\text{coin}_1, \dots, \text{coin}_d$ that changes money.

Greedy Way

GreedyChange(money)

Change \leftarrow empty collection of coins

while money $>$ 0:

 coin \leftarrow largest denomination

 that does not exceed money

 add coin to Change

 money \leftarrow money - coin

return Change

Changing Money

in the US

$$40 \text{ cents} = 25 + 10 + 5$$

Greedy



Changing Money

in Tanzania

$$40 \text{ cents} = 25 + 10 + 5 = 20 + 20$$

Greedy is **not Optimal**



Recursive Change

Given the denominations 6, 5, and 1, what is the minimum number of coins needed to change 9 cents?

MinNumCoins(9) = ?

Money	1	2	3	4	5	6	7	8	9	10
MinNumCoins									?	

Recursive Change

Given the denominations 6, 5, and 1, what is the minimum number of coins needed to change 9 cents?

$$\text{MinNumCoins}(9) = \min \begin{cases} \text{MinNumCoins}(9 - 6) + 1 \\ \text{MinNumCoins}(9 - 5) + 1 \\ \text{MinNumCoins}(9 - 1) + 1 \end{cases}$$

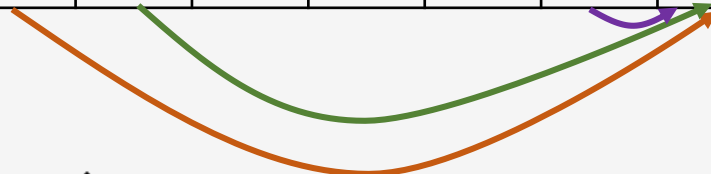
Money	1	2	3	4	5	6	7	8	9	10
MinNumCoins									?	

Recursive Change

Given the denominations 6, 5, and 1, what is the minimum number of coins needed to change 9 cents?

$$\text{MinNumCoins}(9) = \min \begin{cases} \text{MinNumCoins}(3) + 1 \\ \text{MinNumCoins}(4) + 1 \\ \text{MinNumCoins}(8) + 1 \end{cases}$$

Money	1	2	3	4	5	6	7	8	9	10
MinNumCoins									?	

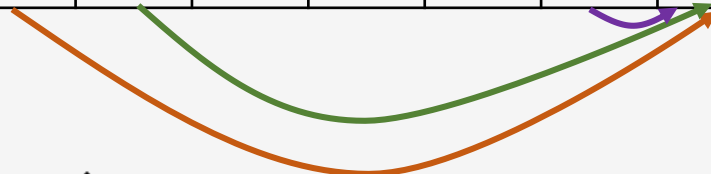


Recursive Change

Given the denominations 6, 5, and 1, what is the minimum number of coins needed to change 9 cents?

$$\text{MinNumCoins}(9) = \min \begin{cases} \text{MinNumCoins}(3) + 1 \\ \text{MinNumCoins}(4) + 1 \\ \text{MinNumCoins}(8) + 1 \end{cases}$$

Money	1	2	3	4	5	6	7	8	9	10
MinNumCoins			?	?				?	?	



Recurrence for Change Problem

MinNumCoins(9) =

$$\min \left\{ \begin{array}{l} \text{MinNumCoins}(\text{money} - \text{coin}_1) + 1 \\ \text{MinNumCoins}(\text{money} - \text{coin}_2) + 1 \\ \dots \\ \text{MinNumCoins}(\text{money} - \text{coin}_n) + 1 \end{array} \right.$$

Recurrence for Change Problem

```
RecursiveChange(money, coins)
```

```
  if money = 0:
```

```
    return 0
```

```
  MinNumCoins  $\leftarrow$   $\infty$ 
```

```
  for i from 1 to |coins|:
```

```
    if money  $\geq$  coini:
```

```
      NumCoins  $\leftarrow$  RecursiveChange(money - coini , coins)
```

```
      if NumCoins + 1 < MinNumCoins:
```

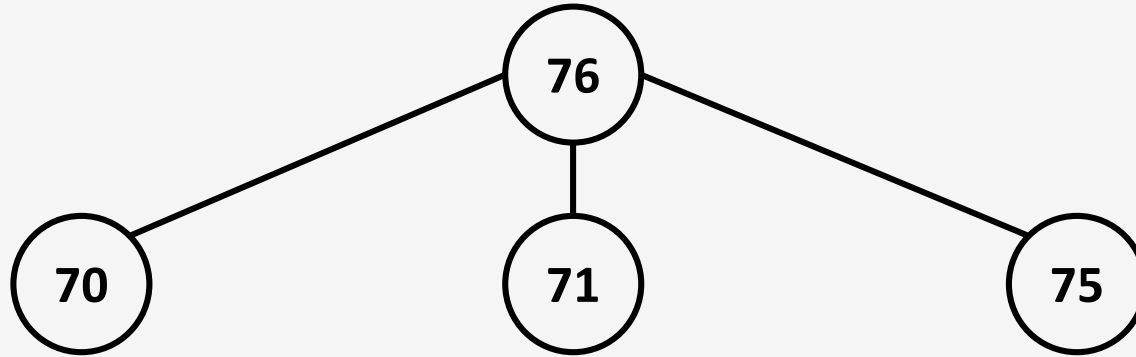
```
        MinNumCoins  $\leftarrow$  NumCoins + 1
```

```
  return MinNumCoins
```

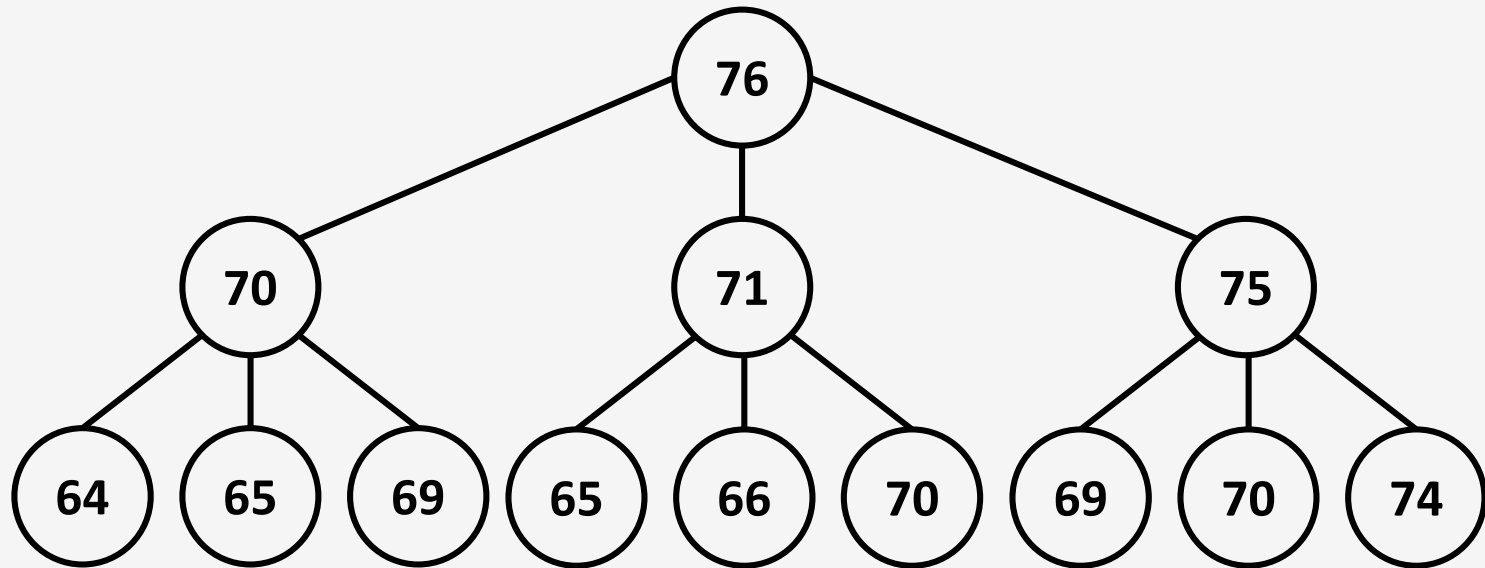
How Fast is RecursiveChange?

76

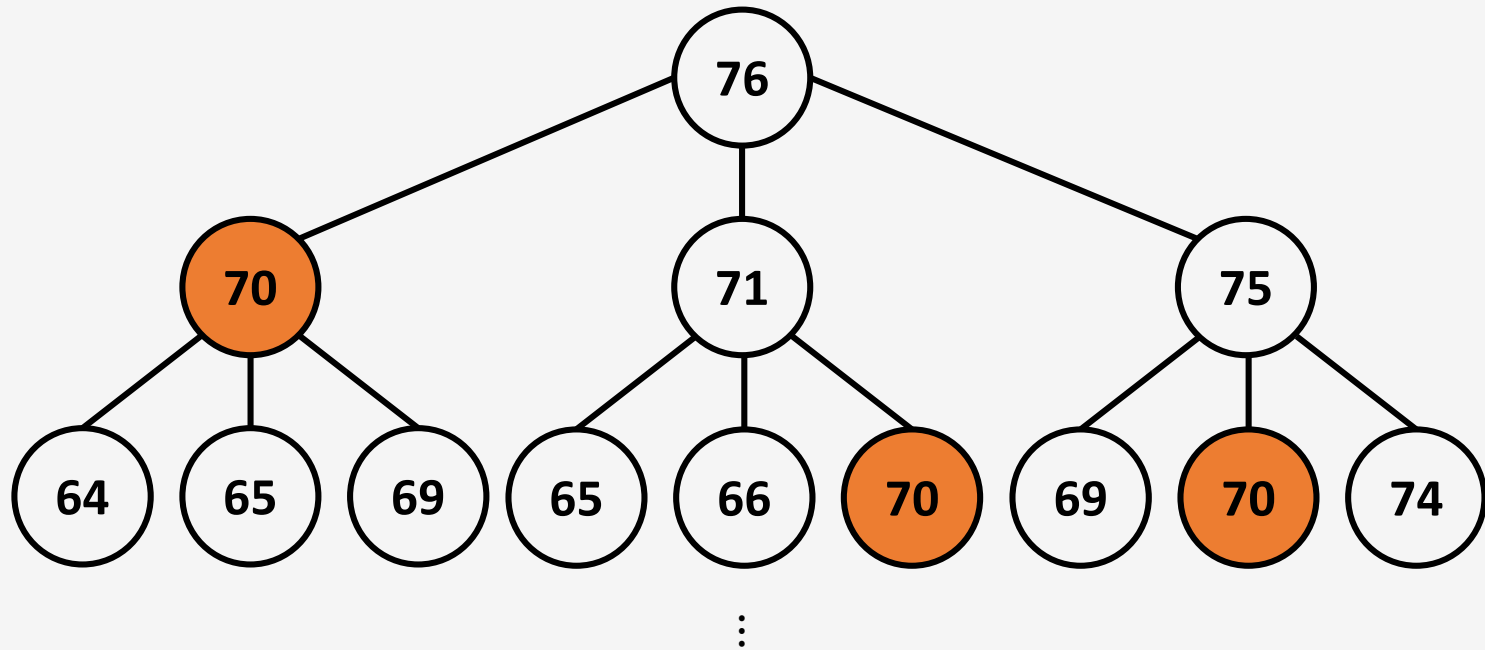
How Fast is RecursiveChange?



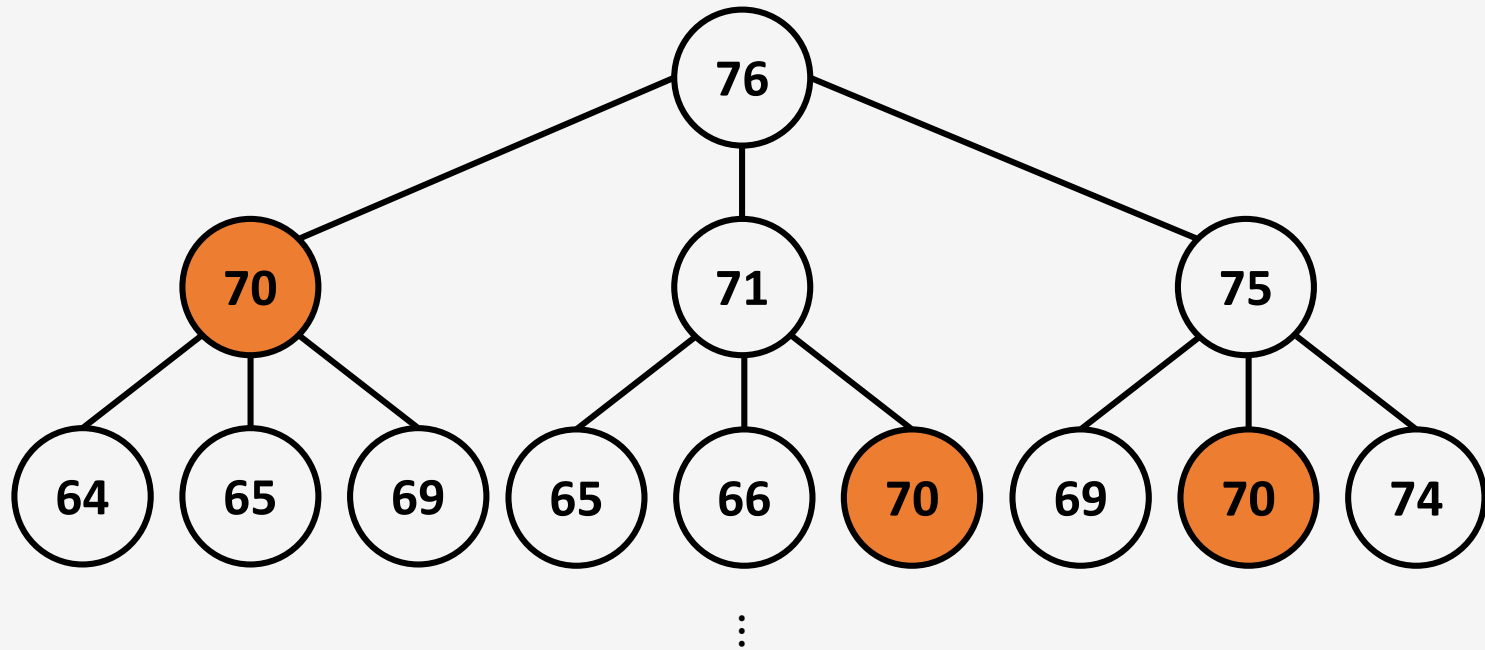
How Fast is RecursiveChange?



How Fast is RecursiveChange?

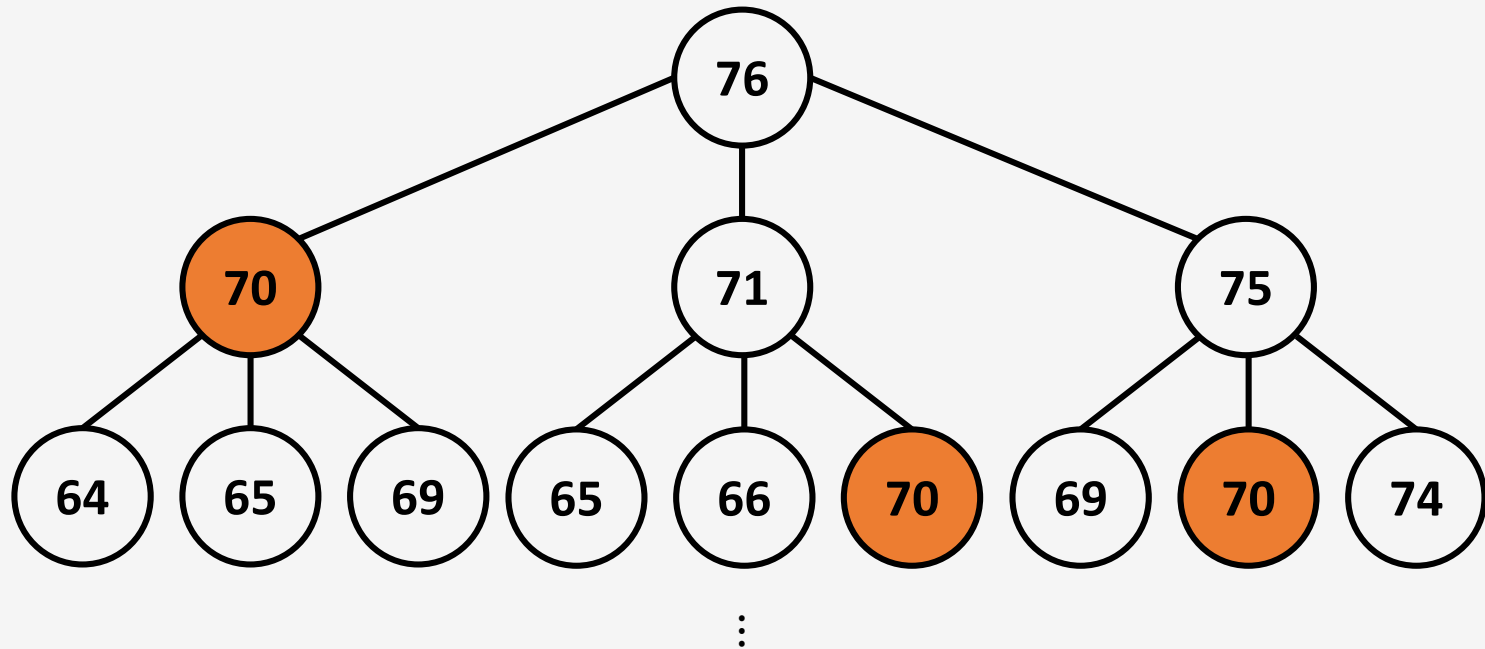


How Fast is RecursiveChange?



the optimal coin combination for 70 cents is computed at least **three** times!

How Fast is RecursiveChange?



the optimal coin combination for 70 cents is computed at least **three** times!

the optimal coin combination for 30 cents is computed at **trillions** of times!

Hint

Wouldn't it be nice to know all the answers for changing **money** – **coin_i** by the time we need to compute an optimal way of changing money?



Hint

Wouldn't it be nice to know all the answers for changing $\text{money} - \text{coin}_i$ by the time we need to compute an optimal way of changing money?

Instead of the time-consuming calls to `RecursiveChange(money-coini, coins)` we would simply look up these values!



Dynamic Programming

Hint

Wouldn't it be nice to know all the answers for changing **money** – **coin_i** by the time we need to compute an optimal way of changing money?

Instead of the time-consuming calls to **RecursiveChange(money – coin_i, coins)** we would simply look up these values!



Dynamic Programming


What is the minimum number of coins needed to change 0 cents for denominations 6, 5, and 1?

Money	0	1	2	3	4	5	6	7	8	9
MinNumCoins	0									

Dynamic Programming

What is the minimum number of coins needed to change 1 cents for denominations 6, 5, and 1?

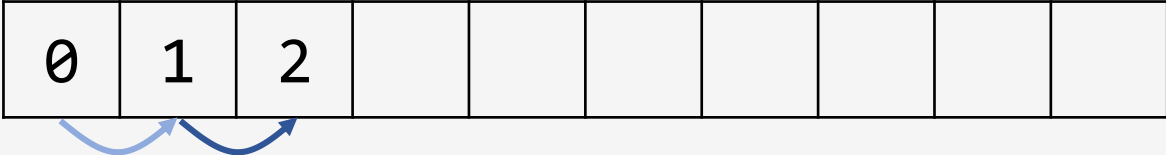
Money	0	1	2	3	4	5	6	7	8	9
MinNumCoins	0	1								



Dynamic Programming

What is the minimum number of coins needed to change 2 cents for denominations 6, 5, and 1?


Money	0	1	2	3	4	5	6	7	8	9
MinNumCoins	0	1	2							



Dynamic Programming

What is the minimum number of coins needed to change 3 cents for denominations 6, 5, and 1?


Money	0	1	2	3	4	5	6	7	8	9
MinNumCoins	0	1	2	3						



Dynamic Programming

What is the minimum number of coins needed to change 4 cents for denominations 6, 5, and 1?

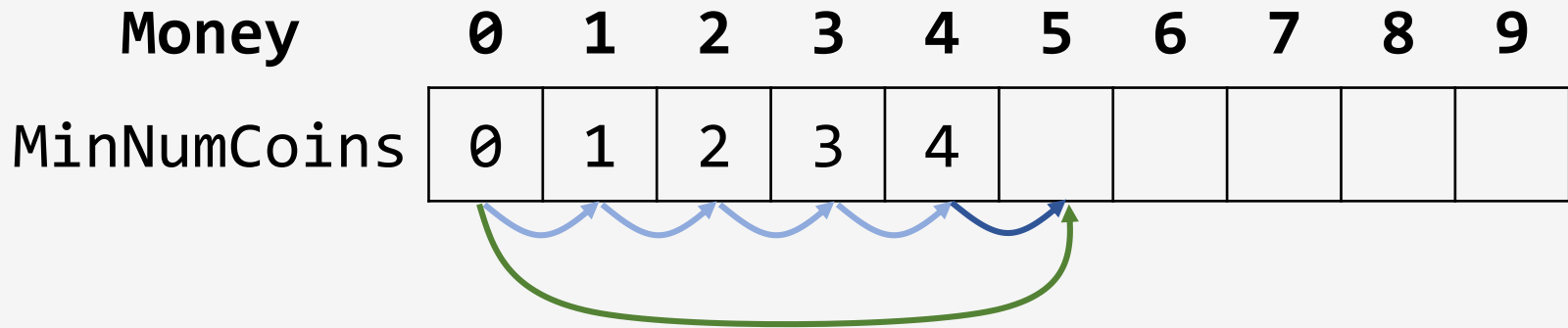
Money	0	1	2	3	4	5	6	7	8	9
MinNumCoins	0	1	2	3	4					



The diagram shows a table with 11 columns representing money values from 0 to 9. The first row is labeled 'Money' and the second row is labeled 'MinNumCoins'. The values in the 'MinNumCoins' row are 0, 1, 2, 3, 4, and then empty cells for 5 through 9. Below the first five cells (0 to 4), there are blue curved arrows pointing from the cell at index i to the cell at index i+1, indicating that the minimum number of coins for amount i+1 depends on the minimum number of coins for amount i.

Dynamic Programming

What is the minimum number of coins needed to change 5 cents for denominations 6, 5, and 1?



$$\min \begin{cases} \text{MinNumCoins}(0) + 1 \\ \text{MinNumCoins}(4) + 1 \end{cases}$$

Dynamic Programming

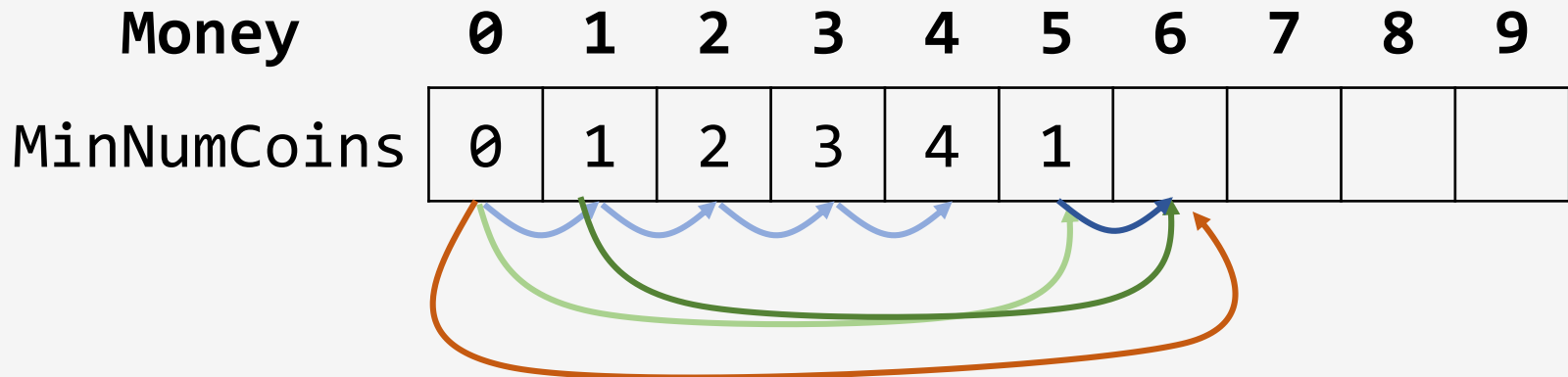
What is the minimum number of coins needed to change 5 cents for denominations 6, 5, and 1?

Money	0	1	2	3	4	5	6	7	8	9
MinNumCoins	0	1	2	3	4	1				

$$\min \begin{cases} \text{MinNumCoins}(0) + 1 \\ \text{MinNumCoins}(4) + 1 \end{cases}$$

Dynamic Programming

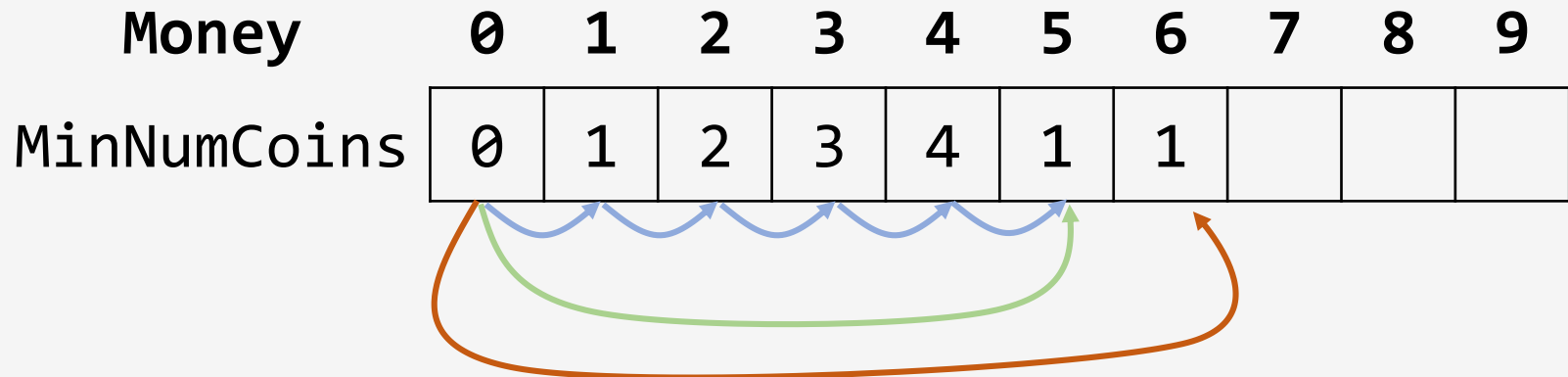
What is the minimum number of coins needed to change 6 cents for denominations 6, 5, and 1?



$$\min \begin{cases} \text{MinNumCoins}(0) + 1 \\ \text{MinNumCoins}(1) + 1 \\ \text{MinNumCoins}(5) + 1 \end{cases}$$

Dynamic Programming

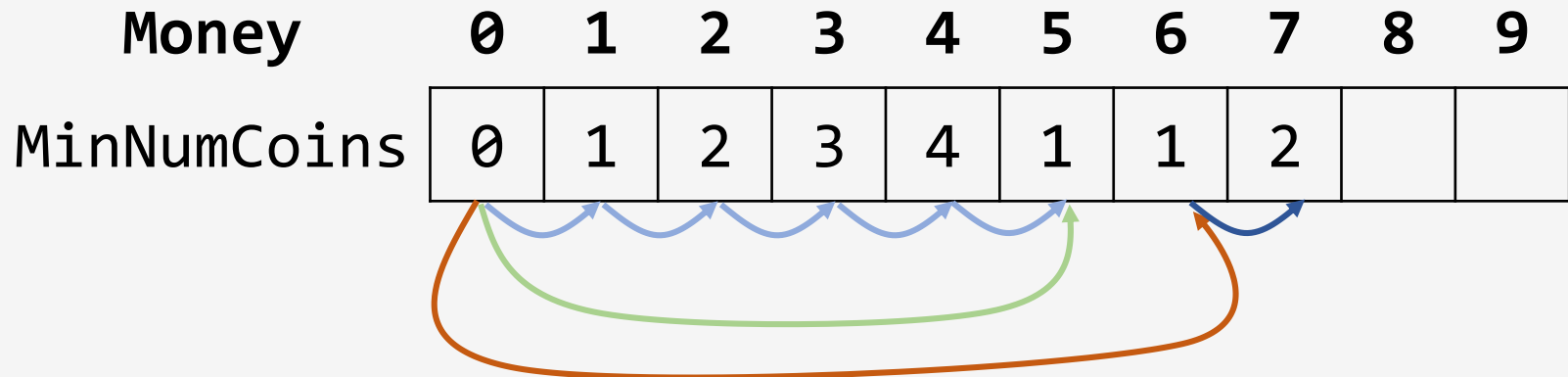
What is the minimum number of coins needed to change 6 cents for denominations 6, 5, and 1?



$$\min \begin{cases} \text{MinNumCoins}(0) + 1 \\ \text{MinNumCoins}(1) + 1 \\ \text{MinNumCoins}(5) + 1 \end{cases}$$

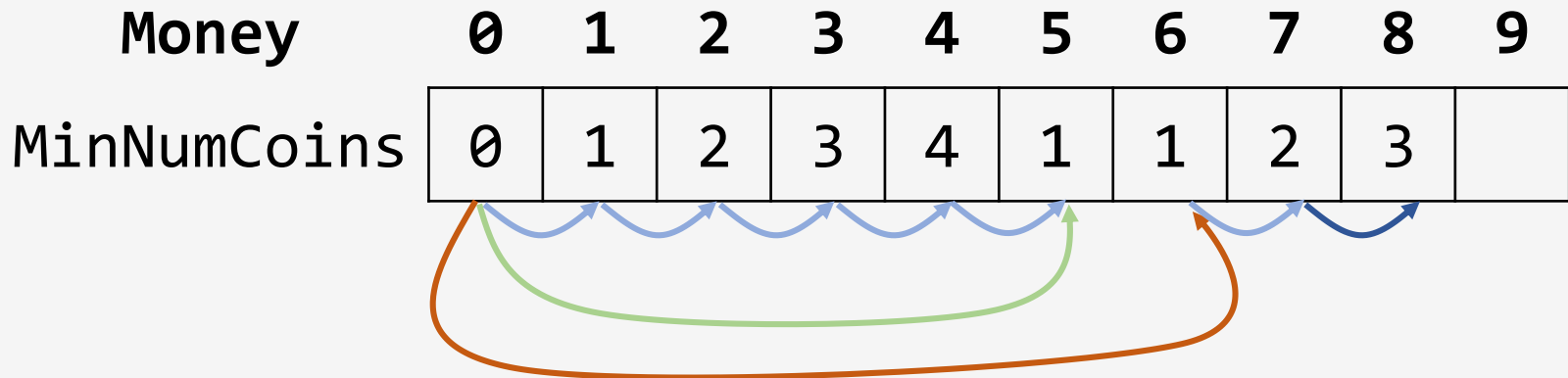
Dynamic Programming

What is the minimum number of coins needed to change 7 cents for denominations 6, 5, and 1?



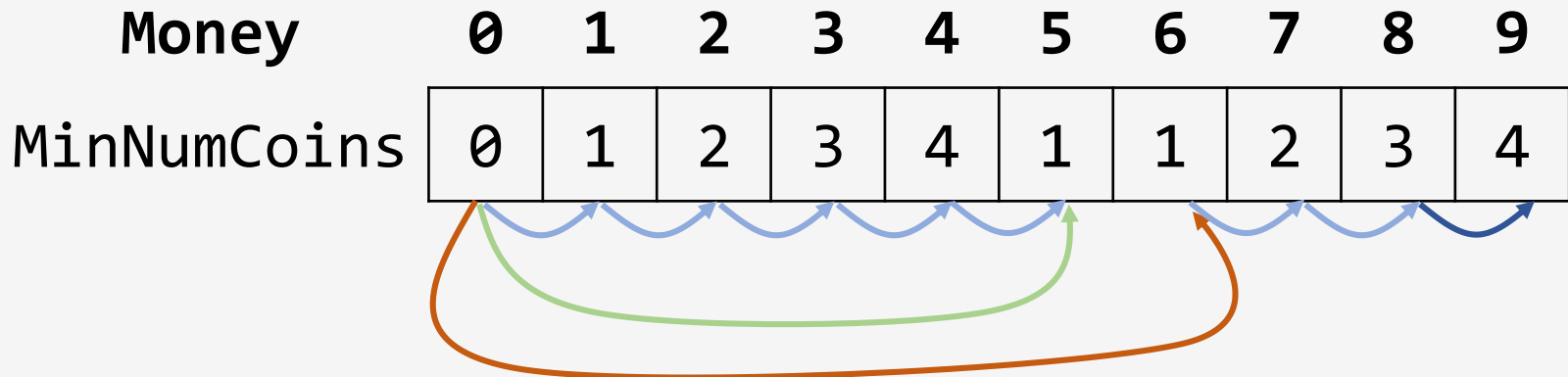
Dynamic Programming

What is the minimum number of coins needed to change 8 cents for denominations 6, 5, and 1?



Dynamic Programming

What is the minimum number of coins needed to change 8 cents for denominations 6, 5, and 1?



Dynamic Solution

DPChange(money, coins)

MinNumCoins(0) \leftarrow 0

for m from 1 to money:

MinNumCoins(m) \leftarrow ∞

for i from 1 to |coins|:

if $m \geq \text{coin}_i$:

NumCoins \leftarrow MinNumCoins(m - coin_i) + 1

if NumCoins < MinNumCoins(m):

MinNumCoins(m) \leftarrow NumCoins

return MinNumCoins(money)

“Programming” in “Dynamic Programming” Has Nothing to Do with Programming!

Richard Bellman developed this idea in 1950s working on an Air Force project.

At that time, his approach seemed completely impractical.

He wanted to hide that he is really doing math from the Secretary of Defense.



Richard Bellman

Richard Bellman

.. What name could I choose? I was interested in planning but planning, is not a good word for various reasons. I decided therefore to use the word, “programming” and I wanted to get across the idea that this was dynamic. It was something not even a Congressman could object to. So I used it as an umbrella for my activities.