

Exercise 1 The coin-change problem consists of the following: Given a set of n coins $\{c_1, c_2, \dots, c_n\}$, each of which is a positive integer, and an amount A , also a positive integer, find whether it is possible to get a change for A . Assume that $A \geq c_i$, $i = 1, \dots, n$.

- a) Consider the two following backtracking algorithms for this problem. Discuss the difference between the two approaches and write the first call.

Function *Coin1*(A, i)

```

if  $A = 0$  then
  return true
if  $A < 0$  or  $i > n$  then
  return false
if Coin1( $A - c_i, i + 1$ ) = true then
  return true
if Coin1( $A, i + 1$ ) = true then
  return true
return false

```

Function *Coin2*(A, i)

```

if  $A = 0$  then
  return true
if  $A < 0$  then
  return false
for  $j = i + 1, \dots, n$  do
  if Coin2( $A - c_j, j$ ) = true then
    return true
return false

```

- b) Draw the recursion call trees that describe the search process of the two algorithms above for the following input data: $c_1 = 3$, $c_2 = 5$, $c_3 = 7$ and $A = 12$; provide only information about the arguments of the recursion call at each node of the tree.
- c) What is the best ordering of the coins with respect to the running time of both backtracking algorithms?
- d) If you want to know the number of possible ways of making change, what needs to be modified in both algorithms?
- e) If you want to know the least number of coins to make the change, what needs to be modified in both algorithms?
- f) What is the time and space complexity of both approaches?

Exercise 2 Read the problem *Zé Manel is setting up a computer network* in EA2022_PL in Mooshak and solve them using a backtracking approach.