1. (6 points) Suppose you’re living in a country with a monetary system in which the coins have values \( \{c_1, \ldots, c_n\} \), where \( c_i \in \mathbb{Z}^+ \), \( 1 \leq i \leq n \). Given \( \{c_1, \ldots, c_n\} \) and \( m \in \mathbb{Z}^+ \), we want to know if it is possible to express \( m \) exactly by drawing coins from an infinite supply of coins of each denomination. For example, if \( \{c_1, \ldots, c_n\} = \{5, 7, 12\} \), then it is not possible to express \( m=16 \) exactly with these coins, but it is possible to express \( m=22 \) exactly with these coins, either as 5, 5, 7, 7 or as 5, 5, 12. Show how to solve this problem for any \( \{c_1, \ldots, c_n\} \) and \( m \) by reducing it to an instance of the KNAPSACK PROBLEM.

2. (4 points) Do Exercise 25.2-6 from pg. 635 of our text.

3. (8 points) A recursive definition of the binomial coefficients is that

\[
\binom{n}{k} = \begin{cases} 
1, & \text{if } k = 0 \vee k = n \\
\binom{n-1}{k-1} + \binom{n-1}{k}, & \text{otherwise}
\end{cases}
\]

The corresponding recursive program is

\[
\text{if } k = 0 \vee k = n \\
\quad \text{then return } 1 \\
\quad \text{else return } \binom{n-1}{k-1} + \binom{n-1}{k}
\]

a What is the time to execute \( \binom{n}{k} \), as a function of \( n \) and \( k \)?

b Write a dynamic programming algorithm to compute \( \binom{n}{k} \) with an execution time in \( \Theta(nk) \).
1. For each coin \( c_i \), we make \( \left\lfloor \frac{W}{c_i} \right\rfloor \) copies of the coin. Choosing \( W=m \) and 
\[ v_{i,j} = w_{i,j} = c_i, \ 1 \leq i \leq n, \ 1 \leq j \leq \left\lfloor \frac{W}{c_i} \right\rfloor, \] we note that it is possible to express \( m \) exactly with the coins if and only if there is a packing of the knapsack of value \( m=W \).

2. The graph admits a negative length cycle if and only if there is an \( i, \ 1 \leq i \leq n \), such that \( d_i^{(n)} < 0 \).

\[
\text{negativecycle} \leftarrow \text{false} \\
\text{for } i \leftarrow 1 \text{ to } n \text{ do} \\
\quad \text{if } d_i^{(n)} < 0 \text{ then } \text{negativecycle} \leftarrow \text{true} \\
\text{return negativecycle}
\]

3. \( a \) Because ultimately the recursion tree causes \( \binom{n}{k} \) 1’s to be added together, computing 

\[
C(n,k) \text{ makes } \binom{n}{k} \text{ calls on } C, \text{ and has a time complexity } \Theta\left(\binom{n}{k}\right).
\]

\( b \) The values are stored in an array \( C[0..n,0..k] \).

\[
\text{for } i \leftarrow 0 \text{ to } n \text{ do} \\
\quad \text{for } j \leftarrow 0 \text{ to } \min(i,k) \text{ do} \\
\quadquad \text{if } k = 0 \lor k = n \\
\quadquad \quad \text{then } C[i, j] \leftarrow 1 \\
\quadquad \text{else } C[i, j] \leftarrow C[i-1, j-1] + C[i-1, j]
\]