1. Let \( n \) be a positive integer and \( k \) be an integer such that \( 0 \leq k \leq n \). Define \( C(n, k) \) as the number of different subsets of \( \{1, 2 \ldots , n\} \) that have size \( k \). For example, \( C(3, 0) = 1 \) as the only subset of \( \{1, 2, 3\} \) that has size 0 is the subset \( \phi \). Similarly, \( C(3, 3) = 3 \) because there are three different subsets of \( \{1, 2, 3\} \) (namely \( \{1, 2\} \), \( \{1, 3\} \) and \( \{2, 3\} \)) that have size 2. Enumerate all different subsets of \( \{1, 2, 3, 4, 5\} \) that have size 3. What is the value of \( C(5, 3) \)?

\( 10 \) pts
2. One can count the number of subsets of \( \{1, 2, 3 \ldots, n\} \) that have size \( k \) (where \( 0 \leq k \leq n \)) by thinking recursively. As a base case, when \( k = 0 \), the number of subsets of \( \{1, 2, 3 \ldots, n\} \) that have size 0 is 1 (only the empty set). Also, if \( k = n \) there is exactly one subset of \( \{1, 2, 3 \ldots, n\} \) of size \( k \) (the whole set). Otherwise, any subset \( S \) of size \( k \) is

(i) EITHER of the type \( S' \cup \{n\} \) where \( S' \) is a subset of \( \{1, 2, \ldots n-1\} \) that has size \( k-1 \). (These are the \( k \)-sized subsets of \( \{1, 2, 3 \ldots, n\} \) that have the element \( n \).)

(ii) OR is of the type \( S'' \) where \( S'' \) is a subset of \( \{1, 2, \ldots n-1\} \) that has size \( k \). (These are the \( k \)-sized subsets of \( \{1, 2, 3 \ldots, n\} \) that do not have the element \( n \)).

Use this idea to derive a recurrence relation for \( C(n, k) \).

15 pts
3. Use the recurrence in part 2 to give a simple recursive procedure \texttt{CHOOSE}(n, k) that computes and returns the value \(C(n, k)\). Draw the complete tree depicting all the recursive calls when a call \texttt{CHOOSE}(5, 3) is made. Indicate in the tree some of the computation that is repeated.

15 pts
4. Give pseudocode for an dynamic programming scheme that avoids the recomputation of the simple recursive procedure to efficiently calculate $C(n, k)$. Your scheme must use memoization to efficiently calculate $C(n, k)$ using the recurrence relation obtained in part 2. What is the running time of your procedure?

20 pts