Program Documentation

Language: Java

Main Java File: Question1.java

Main Class name: Question1

**Description**:

* The problem is defined as follows: given a set of items "U" and a set of cover sets "v", find the minimum number of cover sets from "v" that together cover all items in "U".
* The U set is initialized to contain integers from 1 to n.
* The code uses a backtracking approach to find all the possible combinations of sets in v that covers every element in whole set U.

**Reading Input:**

* The program uses a Scanner object to read input from the user.
* It reads the input as two integers (n and m), followed by "m" lines of strings.
* Each line of string represents a key and its associated set of integers. The key’ key[1].substring(0,key[1].indexOf(":"))’ is stored as the key of the HashMap and its associated set”v1” is stored as the value of the HashMap.

**Processing the Input:**

* The program creates a Set "v12" to store solutions in the form of ArrayLists.
* It calls a recursive function "xx" with arguments (p, i, v, C, U, sol, v12) where "p" is the starting position and "i" is the ending position of the set being processed.
* The function "xx" uses recursive backtracking Algorithm to find a solution.
* The method xx implements the backtracking. It takes 7 arguments:
  + p: the starting index of the set to be considered in the current iteration
  + i: the maximum number of sets that can be combined
  + v: the HashMap storing the sets of integers
  + C: the set containing the elements that are covered so far in the current iteration
  + U: the set containing all the elements that need to be covered
  + sol: an ArrayList that stores the sets that are combined so far in the current iteration
  + v12: a set that stores the solutions that have already been found to avoid counting them again
* The xx method uses a for loop to combine sets from the v map and adds the elements of each set to C set.
* If the C set is equal to the U set, the solution is considered as an optimal solution and this is the base condition for the recursion.
* The solution is stored in the v12 set to avoid counting it again.
* The code prints the total number of optimal solutions found.

**Final Output:**

* If no solution is found, the program prints "No solutions".
* If one or more solutions are found, the program prints the number of optimal solutions.