Oberon Ilano

CS 430

Sum of Subset using Backtracking

December 2, 2019

The purpose of this program is to find all the possible combinations of the integers that sum to S. The expected inputs from the user are the values for S, n, and array X. The value for S is inputted from the keyboard to be compared to all possible combinations of the integers. The value for n is inputted from the keyboard for the size of the array X. The n integers will be inputted from the keyboard to be stored in the array X and to be compared for all possible combinations that will sum to S.

Example 1

*Input:*

11 for S

7 for n

6, 9, 8, 4, 7, 5, and 2 for integers in array X with size n

Enter the target sum value S: **11**

Enter how many integers are in the set: **7**

Enter 7 positive integers: **6 9 8 4 7 5 2**

*Output:*

7 integers that are given in nondecreasing order:

**2 4 5 6 7 8 9**

Set of solution(s) that sum to 11:

**{ 2 4 5 }**

**{ 2 9 }**

**{ 4 7 }**

**{ 5 6 }**

Example 2

*Input:*

9 for S

5 for n

6, 5, 7, 11, and 2 for integers in array X with size n

Enter the target sum value S: **9**

Enter how many integers are in the set: **5**

Enter 5 positive integers: **6 5 7 11 2**

*Output:*

5 integers that are given in nondecreasing order:

**2 5 6 7 11**

Set of solution(s) that sum to 9:

**{ 2 7 }**

Example 3

*Input:*

7 for S

5 for n

5, 9, 12, 3, and 8 for integers in array X with size n

Enter the target sum value S: **7**

Enter how many integers are in the set: **5**

Enter 5 positive integers: **5 9 12 3 8**

*Output:*

5 integers that are given in nondecreasing order:

**3 5 8 9 12**

Set of solution(s) that sum to 7: **NO SOLUTION FOUND!**