5.BackTracking

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#define MAX\_ELEMENTS 100

void printSubset(int subset[], int subsetSize) {

for (int i = 0; i < subsetSize; i++) {

printf("%d ", subset[i]);

}

printf("\n");

}

void subsetSumBacktrack(int set[],int n,int subset[],int subsetSize,int sum,int targetSum,int currentIndex)

{

if(sum == targetSum)

{

printSubset(subset,subsetSize);

return;

}

if (currentIndex >= n || sum > targetSum) {

return;

}

subset[subsetSize]=set[currentIndex];

subsetSumBacktrack(set,n,subset,subsetSize+1,sum+set[currentIndex],targetSum,currentIndex + 1);

subsetSumBacktrack(set,n,subset,subsetSize,sum,targetSum,currentIndex+1);

}

void solveSubsetSum(int set[], int n, int targetSum) {

int subset[MAX\_ELEMENTS];

subsetSumBacktrack(set,n,subset,0,0,targetSum,0);

}

int main()

{

int n;

printf("Enter the number of elements in the set: ");

if (scanf("%d",&n)!=1 || n <= 0 || n > MAX\_ELEMENTS)

{

printf("Invalid input for the number of elements.\n");

exit(1);

}

int set[MAX\_ELEMENTS];

printf("Enter the elements of the set:\n");

for (int i = 0; i < n; i++) {

if (scanf("%d", &set[i]) != 1) {

printf("Invalid input for element \n", i);

exit(1);

}

}

int targetSum;

printf("Enter the target sum: ");

if (scanf("%d", &targetSum) != 1) {

printf("Invalid input for the target sum.\n");

exit(1);

}

printf("Subsets with sum equal to %d are:\n",targetSum);

solveSubsetSum(set,n,targetSum);

return 0;

}

OUTPUT



