International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 6, Issue 6, November- December 2017 ISSN 2278-6856

Subset Sum Problem-New Representation Approach for finding the solution

Dr. Anil Kumar Singh[1], Mr. Anand Kumar Dixit[2]

[1]Jagran Institute of Management 620-W block Saket Nagar, Kanpur-208014 (U.P.), India

[2]Jagran Institute of Management 620-W block Saket Nagar, Kanpur-208014 (U.P.), India

Abstract

The Subset Sum problem is an non deterministic problem where we need to find a resultant number from a set of numbers to perform the addition of subset of a set. Non-polynomial problems comprise of the set of decision problems where answer to any instance of the problem is true then it can be easily proved why the solution is true. Non-Deterministic Polynomial problems are the collection of problems where if the solution is true, then it provides the complexity of the problem in polynomial time [1]. In this paper we will demonstrate the new representation method for subset sum problem which is very easy and help to the students and research scholar to find the solution.

Keywords: Subset Sum Problem, Non-Deterministic Polynomial problem (NPC), Password generation techniques, Backtracking.

1.INTRODUCTION

In the subset sum problem we have to find a subset s` of the given set S=<S1,S2,S3……$S_n$> where the elements of the set S are n positive integers in such a manner that s` ϵ S and the sum of the elements of subset s` is equal to some positive integer 'X'. It is well known that the above problem is NP complete; there is trivial solution with computational complexity $O(2^n)$ [2]. The sub set problem can be solved by using various approaches like dynamic techniques, backtracking approaches etc.

Dynamic programming is a useful technique for making a sequence of interrelated decisions. It offers a step wise technique for finding the optimal combination of decisions. The multistage decision policy with recursive approach will provides an efficient way while using Dynamic programming. In the scenario of multistage decision process the problem is divided into several parts known as sub problems, then each and every sub problem will be resolved individually and the end result will be found by combining the results of all the sub problems [3]. It is a difficult and complex method. Now we take another approach.

2. BACKTRACKING & ITS REPRESENTATION

Backtracking approach is also used for the solution of above problem. The power of backtracking comes into existence when we combine explicit and implicit constraints, and we stop generating nodes when these constraints fail. We can improve the above algorithm by strengthening the constraint checks. In this approach we create a implicit tree for solution. This tree is binary tree. The root of the tree is selected in such a way that its represent that no decision is yet taken on any input. We assume that the elements of the given set are arranged in increasing order: S1≤S2≤S3……≤S_n The left child of the root indicates that we have to include S1 from the set S and the right child of the root node indicates that we have to exclude S1. Each node stores the sum of the partial solution elements. If at any stage the number equals to X then the search is successful and terminates.

The dead end in the tree occurs only when either of the two inequalities exists [4]:
The sum of s` is too large: s`+S_i+1>X.
The sum of s` is too small: s`+S_i+1<X
Let us take an example: A sequence of set is given by S= <1,3,4,5> and X=8. Solution of above problem is given by an implicit tree.

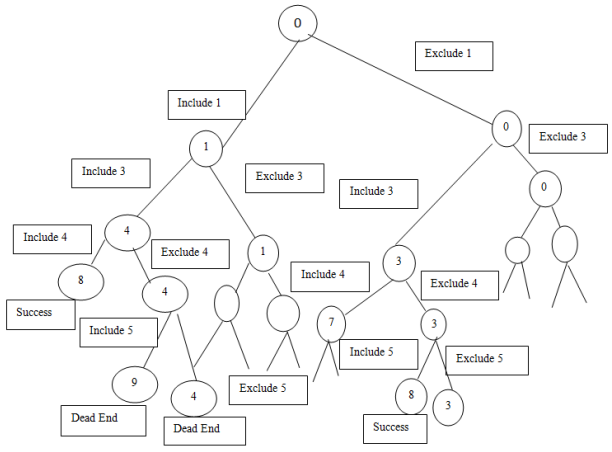

Fig. Implicit tree partial representation

In above implicit tree there are various branches which are used to find answer. So it seems to complex

3.MODIFIED METHOD & IT`S ALGORITHM

We assume that the elements of the given set are arranged in increasing order: $S1 \leq S2 \leq S3 \ldots\ldots\ldots \leq S_n$ so algorithm for modified process is given below:

Step1. First we take a pair of two numbers from sequence set and find the addition of all pairs. On the basis of addition, we will take the decision that which of the pairs will be goes to next round.

Step 2. if (sum of pair == X) then
print "SS: Search Successful "
else if (sum of pair < X) then
goto Next Round and print "NR"
else if (sum of pair < X && one of the element is last element of set)
print "DE: Dead End "

Step 3. We will increase a number in pairs and perform step1 and step2.

Step 4. The above process will continue until all decision status become Dead End. Now we are taking two examples to implement the new modified algorithm.

Step 1:

Pair Set	1,3	1,4	1,5	3,4	**3,5**	4,5
Addition	4	5	6	7	**8**	9
Decision	NR	NR	DE	NR	**SS**	DE

Step 2:

Pair Set	**1,3,4**	1,3,5	1,4,5	3,4,5
Addition	8	9	10	12
Decision	SS	DE	DE	DE

Hence all the solutions are gained:

R1=<3,5> and R2=<1,3,4>

Example2.

Given S= <1, 2, 3, 4, 5, 6> and X=9.
We can represent through matrix format rather than tree format because it is difficult.
Step 1:

Step 2:

Pair Set	1,2	1,3	1,4	1,5	1,6	2,3
Addition	3	4	5	6	7	5
Decision	NR	NR	NR	NR	DE	NR

Pair Set	2,4	2,5	2,6	3,4	3,5	3,6
Addition	6	7	8	7	8	9
Decision	NR	NR	DE	NR	NR	SS
Pair Set	4,5	4,6	5,6			
Addition	9	10	11			
Decision	SS	DE	DE			

Step 3:

Pair Set	1,2,3	1,2,4	1,2,5	**1,2,6**	1,3,4	1,3,5
Addition	6	7	8	**9**	8	9
Decision	NR	NR	NR	**SS**	NR	SC
Pair Set	1,3,6	1,4,5	1,4,6	1,5,6	**2,3,4**	2,3,5
Addition	10	10	11	12	**9**	10
Decision	DE	DE	DE	DE	**SS**	DE
Pair Set	2,3,6	2,4,5	2,4,6	2,5,6	3,4,5	3,4,6
Addition	11	11	12	13	12	13
Decision	DE	DE	DE	DE	DE	DE
Pair Set	3,5,6					
Addition	14					
Decision	DE					

Pair Set	1,2,3,4	1,2,3,5	1,2,3,6	1,2,4,5
Addition	10	11	12	12
Decision	DE	DE	DE	DE
Pair Set	1,2,4,6	1,2,5,6	1,3,4,5	1,3,4,6
Addition	13	14	13	14
Decision	DE	DE	DE	DE

5.USES OF THE SUBSET SUM PROBLEM

It is only the specially constructed problems which are known to be easy. There are security problems other than public-key codes for which subset-sum problems are useful. We can use it to computer security. With the help of this we can generate an alternate for system password. Message verification is also done by this subset sum.

5.1 Computer passwords

A PC mandatory to verify a user's identity before allowing him/her access to an account. The simplest system would have the computer store a password's copy in an internal database file, and compare it with what the user enters at time of login. A problem is that anyone who sees the internal database file could later imitate the user.
It is found that this alternative solution is essentially implemented on the base of subset sum system: the PC produces a large set of numbers, like six hundred numbers, of ai. These are stored in the internal database file. A password is a subset of {1….600}. Instead of having the password for the user, the computer stores the sum of associated numbers with the suitable subset. When the user enters in the subset, the computer verifies whether the sum of numbers is equal to desired total. It does not store a record of the subset. Therefore, impersonation is possible

only if any user can rebuild the subset knowing the a_i and the total.

5.2 Message verification

A sender wants to send messages to a receiver. Keeping the message secret is important. So receiver wants to be sure that the message he/she is receiving is not from a fraud and has not been tampered. So with the help of subset sum problem we can generate a verification code so that anybody who wants to read, first verify this by a password. Password is generated by above technique [5].

6.CONCLUSION

It is found that the above proposed representation technique is easy to implicit tree technique. With the help of this method we can easily represent and find the solution. It is very easy to understand. It is very fast in finding the solution due to the not computing unnecessary pairs value.

References

[1]. M. Alekhnovich, A. Borodin, J. Buresh-Oppenheim, R. Impagliazzo, A. Magen, and T. Pitassi, Toward a Model for Backtracking and Dynamic Programming, Proceedings of the 20th Annual IEEE Conference on Computational Complexity (2005), pp. 308-322.

[2]. N. Alon and G. Freiman, On Sums of Subsets of a Set of Integers, Combinatorica 8:4 (1988), pp. 297-306.

[3]. Anand Kumar Dixit, M. Jain, A. Srivastava, S. Ghosh an initiation of Dynamic Programming to solve the Graphical as well as Network Problems for the minimum path between node. published in "OJCST (International Journal ISSN: 0974-6471) Volume 4 No. 1 225-227 June 2011

[4]. Udit Agarwal, Algorithms design and analysis (2011) DRCo.Ltd. Publication

[5]. http://www.math.stonybrook.edu/~scott/blair/ Other_uses_subset_sum.html

AUTHORS

Dr. Singh has 17 years' experience in Academic and Research. He is a Post Graduate in Computer Science and an MCA. He has received the Ph.D. degree in Information Technology from Mahatama Gandhi Chitrakoot Gramodaya Vishwavidyalaya, Chitrakoot in 2012. He has to his credit several research papers published in National and International journals and conferences. Presently he is working in Jagran Institute of Management, Kanpur as Associate Professor.

Mr. Dixit has 10 years' experience in Academic. He is a Post Graduate in Computer Application. He has to his credit several research papers published and presented in National and International journals and conferences. Presently he is working in Jagran Institute of Management, Kanpur as Assistant Professor.