# Polynomial time algorithm for a sub-problem of SUBSET SUM with exponentially growing input

**Tatiana Kosovskaya**

***Abstract:*** *A polynomial time algorithm for a sub-problem of a well-known NP-complete problem SUBSET SUM is proposed in the paper. The input data of such a sub-problem must satisfy a condition: if numeric parameters are written as an increasing sequence then every parameter value is not less then the sum of all previous ones. Such a condition is fulfilled, for example, for geometric progression $a^i$ ($i = 0, \dots, n$) and number of placements $A_n^i$ ($i = 0, \dots, n$). The algorithm may be easily modified for the sequence of binomial coefficients $C_n^i$ ($i = 0, \dots, [\frac{n+1}{3}], n - [\frac{n+1}{3}], \dots, n$).*

***Keywords:*** *NP-completeness, SUBSET SUM, polynomial time sub-problem*

***ACM Classification Keywords****: F.1.3 Complexity Measures and Classes, Reducibility and completeness; F.2.1 Analysis of Algorithms and Problem Complexity, Numerical Algorithms and Problems*

## Introduction

A pseudo-polynomial algorithm solving the NP-complete problem SUBSET SUM is presented in Garey, Johnson, [1979].

### SUBSET SUM

Instance: A set of positive integers $\{s_1, \dots, s_n\}$ and a positive integer $B$.

Question: If there exists such a subset of $\{s_1, \dots, s_n\}$ that the sum of its elements equals to $B$? That is if there exists a $\{0, 1\}$-solution of the equation $s_1x_1 + \dots + s_nx_n = B$?

The pseudo-polynomial algorithm solving this problem is effective enough for an input where the values of number parameters do not essentially differ from each other. But it is not effective (becomes essentially exponential time) if values of number parameters exponentially differ from each other.

Bellow the following restriction of the problem SUBSET SUM is under consideration. If numeric parameters are written as an increasing sequence then every parameter value is not less then the sum of all previous ones. That is $s_1 < \dots < s_k$, $\sum_{i=1}^{j-1} s_i < s_j \le s$ for every $j = 2, \dots, k$ and $\sum_{i=1}^{k} s_i \ge s$ .

## Polynomial time algorithm for the sub-problem under consideration

Let $S$ be the difference between $s$ and the current value of the sum of the chosen sub-set elements, $l$ be the number of the maximal element which is not less then $S$, $a_1, \dots, a_k$ be $\{0, 1\}$-values of coefficients in linear combination representing $s$ by means of $s_1, \dots, s_n$.

1. $S := s$;

   $a_1, \dots, a_k := 0$;

   $l := \arg\max\{j : s_j \le S\}$.

2. $a_l := 1$;

   $M := l - 1$.

3. $S := S - s_l$;
   $l := \arg\max\{j : s_j \leq S\}$;
   *if* $\{l$ is not defined$\}$ *then* {solution does not exist}.
4. *if* $\{l < M\}$ *then* $\{M := l.\}$
5. $a_M := 1$;
   *if* $\{S = 0\}$ *then* {solution is received} *else* $\{M := l - 1$, *goto* item 3}.

The described algorithm is a recursive one. During every step of recursion one solves the same problem with $s = S$.

The correctness of the algorithm run is provided by the following theorem.

**Theorem 1.** *If a number $s$ is represented as a sum of numbers from an increasing sequence $\{s_1, \ldots, s_k\}$ satisfying conditions $\sum_{i=1}^{j-1} s_i < s_j \leq s$ ($j = 2, \ldots, k$) and $\sum_{i=1}^{k} s_i \geq s$, then $s$ contains the number $s_k$ in this sum.*

Proof. Let the representation does no contain the number $s_k$. Then $s = a_1 s_1 + \cdots + a_{k-1} s_{k-1} < s_k \leq s$ for some coefficients $a_1, \ldots, a_k$ with values from {0, 1}. We come to a contradiction. □

Let's estimate the number of steps of this algorithm. A multi-tape Turing machine will be used as a model for such an estimation.

The recursion depth of the algorithm is not greater then $k$. Remind that $k$ is not an input of the problem and is less then the length of input $k < ||s_1|| + \cdots + ||s_k|| + ||s||$. Moreover, as $\sum_{i=1}^{k-1} s_i < s$ then $||s|| \geq ||\sum_{i=1}^{k-1} s_i|| \geq ||s_1|| + k - 1$. Hence, $k \leq ||s|| - ||s_1|| + 1$ and $O(k) = O(||s||)$.

Items 1 and 2 of the algorithm are fulfilled only once and require $k+3$ assignments and not more than $k$ comparisons of two integers the length of which is not more than the length of $s$. Totally $O(||s||k) = O(||s||^2)$ steps.

Items 3 — 5 are fulfilled not more then $k$ times and require respectively:

– item 3: one subtraction and not more than $k$ comparisons of two integers the length of which is not more than the length of $s$; totally $O(||s|| + ||s||k) = O(||s||^2)$ steps;

– item 4: one comparison of two integers the length of which is not more than the length of $s$ and may be one assignment; $O(||s||)$ steps;

– item 5: one assignment of 1; one comparison with zero and one assignment; totally $O(k) = O(||s||)$ steps.

The number of the algorithm steps realized by a multi-tape Turing machine is $O(||s||^2) + k(O(||s||^2) + O(||s||) + O(||s||)) = O(||s||^3)$ steps. So the following theorem is proved.

**Theorem 2.** *A sub-problem of NP-complete problem SUBSET SUM in which numeric parameters satisfy conditions $s_1 < \cdots < s_k$, $\sum_{i=1}^{j-1} s_i < s_j \leq s$ for every $j = 2, \ldots, k$ and $\sum_{i=1}^{k} s_i \geq s$ belongs to the class* **P**.

## Examples of polynomial time sub-problems of SUBSET SUM

**Lemma 1.** $\sum_{i=1}^{k-1} a^i < a^k$ *for every $k > 1$ and $a \geq 2$.*

Proof. $\sum_{i=1}^{k-1} a^i = \frac{a^k - 1}{a - 1} < \frac{a^k}{a-1} \leq a^k$. □

**Corollary 1 of theorem 2.** *A sub-problem of the problem SUBSET SUM in which $s_1, \ldots, s_k$ are the members of increasing geometric progression belongs to the class* **P**.

**Lemma 2.** *For every $k$ ($k > 1$) and $j$ ($1 \leq j \leq k - 1$) it is valid $\sum_{i=0}^{j-1} A_k^i < A_k^j$.*

Proof (induction on $l$).

Let $j = 1$. Then $A_k^1 = k > 1 = \sum_{j=o}^{0} A_k^j$.

Let $j > 1$ and $A_k^{j-1} > \sum_{i=o}^{j-2} A_k^i$. Prove that $A_k^j > \sum_{i=o}^{j-1} A_k^i$ if $k > 1$ and $j \leq k - 1$.

$$A_k^j = A_k^{j-1}(k - j + 1) = A_k^j + A_k^{j-1}(k - j) >_{\text{(induction hypothesis)}}$$

$$A_k^{j-1} + (k-j)\sum_{i=o}^{j-2} A_k^i \geq \sum_{i=o}^{j-1} A_k^i.$$

The last inequality is valid if $k - j \geq 1$, i.e. if $j \leq k - 1$. □

**Corollary 2 of theorem 2.** *A sub-problem of the problem SUBSET SUM in which $s_i = A_k^i$ ($i = 0, \ldots, k-1$) belongs to the class* **P**.

## Modification of the algorithm for binomial coefficients as input data for SUBSET SUM

Note that the values of binomial coefficients $C_k^l$ increase only for $0 \leq l \leq [\frac{k}{2}]$. These values are symmetrical according the index $[\frac{k}{2}]$ (or the pair of indexes $[\frac{k}{2}]$ and $[\frac{k}{2}] + 1$).

So, we have the input $C_k^0 = C_k^k < C_k^1 = C_k^{k-1} < \cdots < C_k^j = C_k^{k-j} < \cdots < C_k^L = C_k^{k-L}$, where $L = [\frac{k}{2}]$ and $C_k^{k-L}$ is absent if $k$ is even.

Besides, the binomial coefficients does not satisfy the condition $C_k^l > \sum_{j=o}^{l-1} C_k^j$ for every $l$.

**Lemma 4.** *For every $k$ ($k > 1$) and every $l$ ($1 \leq l \leq [\frac{k+1}{3}]$)*

$$C_k^l > \sum_{j=o}^{l-1} C_k^j.$$

Proof (induction on $l$).

Let $l = 1$. Then $C_k^1 = k > \sum_{j=o}^{0} C_k^j = C_k^0 = 1$.

Let $C_k^l > \sum_{j=o}^{l-1} C_k^j$. Prove that $C_k^{l+1} > \sum_{j=o}^{l} C_k^j$ for $k > 1$ and $1 \leq l+1 \leq [\frac{k+1}{3}]$, i.e. $1 \leq l \leq [\frac{k-2}{3}]$.

$$C_k^{l+1} = \frac{k!}{(l+1)!(k-l-1)!} = \frac{k!}{l!(k-l)!}\,\frac{k-l}{l+1} =$$

$$C_k^l\,\frac{k-l}{l+1} = C_k + C_k^l(\frac{k-l}{l+1} - 1) >_{\text{(inductive hypothesis)}}$$

$$C_k^l + (\frac{k-2l-1}{l+1})\sum_{j=o}^{l} C_k^j \geq \sum_{j=o}^{l} C_k^j.$$

The last inequality is valid for $k - 2l - 1 \geq l + 1$, i.e. for $l \leq \frac{k-2}{3}$. □

The proved lemma provides the possibility of modification the above described algorithm for binomial coefficients (with $1 \leq l \leq [\frac{k+1}{3}]$) as input data for SUBSET SUM. The necessity of modification follows from the fact that during every recursion step there may be two equal maximal elements ($l$-th and $(k-l)$-th) for which $S \leq \sigma_l C_k^l$ ($\sigma_l = 1$ or $\sigma_l = 2$).

Let $S$ be the difference between $s$ and the current value of the sum of the chosen sub-set elements, $l$ be the number of the minimal element the sum of which with the less ones is greater then $S$, $a_0, \ldots, a_k$ be $\{0, 1\}$-values of coefficients in linear combination representing $s$ by means of binomial coefficients, $\sigma_i$ be the number of binomial coefficients which are equal to $C_k^i$ and are not included into the representation of $s$, $L = [\frac{k+1}{3}]$.

1. $S := s$;

   $a_0, \ldots, a_k := 0$;

   $\sigma_1, \ldots, \sigma_L := 2$;

   $l := \arg\min\{j : j \leq L \,\&\, S \leq \sum_{i=0}^{j-1} \sigma_i C_k^i\}$.

2. *if* $\{\sigma_l = 2\}$ *then* $\{a_l := 1\}$ *else* $\{a_{k-l} := 1\}$;
   $S := S - C_k^l$;
3. *if* $\{ S = 0\}$ *then* { solution is received }
4. *if* $\{ a_l \neq 0\}$ *then* $\{ \sigma_l := 1\}$;
   $l := \arg\min\{j : j \leq L \,\&\, S \leq \sum_{i=0}^{j-1} \sigma_i C_k^i\}$;
   *if* { $l$ is not defined } *then* { solution does not exist} *else* {goto item 2}.

The described algorithm is a recursive one. During every step of recursion one solves the same problem with $s = S$.

The correctness of the algorithm run is provided by the following theorem.

**Theorem 3.** *Every integer $s$ represented as a sum of some binomial coefficients $C_k^i$ $(i = 0, \ldots, L, k-L, \ldots, k)$ contains in such a representation the number $C_k^l$, $l$ be a maximal index such that $2\sum_{i=0}^{l-1} C_k^i < s$.*

Proof. Let such a representation does not contain the number $C_k^l$.
Then for some coefficients $a_0, \ldots, a_{l-1}, a_k, \ldots, a_{k-l+1}$ from {0, 1} we have
$s = \sum_{i=0}^{l-1} a_i C_k^i + \sum_{i=k}^{k-l+1} a_i C_k^i \leq 2\sum_{i=0}^{l-1} C_k^i < s.$
We come to a contradiction. □

## Conclusion

Essential differences of the well-known NP-complete problem Conjunctive Boolean Query while its implementation as a problem of pattern recognition in the frameworks of logic-objective approach and as a problem of the data base use are shown.

In the both implementations it is possible to construct a level description of a fixed input data which increases the time complexity of multiple implementation. The estimates of number of steps while using a two-level data base are proved in the paper.

Algorithms of level description of classes are yet developed in the previous papers of the author, but there is only an approach to developing of algorithms of level data base construction.

## Acknowledgements

The paper is published with partial support by the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua)

## Bibliography

Garey M.R., Johnson D.S., "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman, New York, 1979.

## Authors' Information

***Tatiana Kosovskaya*** - *Dr., Professor of St.Petersburg State University, University av., 28, Stary Petergof, St.Petersburg, 198504, Russia; e-mail: kosovtm@gmail.com*
*Major Fields of Scientific Research: Logical approach to artificial intelligence problems, Theory of Computational Complexity of Algorithms*