CSE 462 | ALGORITHM ENGINEERING SESSIONAL

The Subset Sum Problem

Group 6

Group Leader - Farhanaz Farheen (1505013)
Sifat Ishmam Parisa(1505016)
Salman Shamil (1505021)
Kazi Sajeed Mehrab (1505025)
Syeda Nahida Akter (1505027)

Bangladesh University of Engineering and Technology

INTRODUCTION

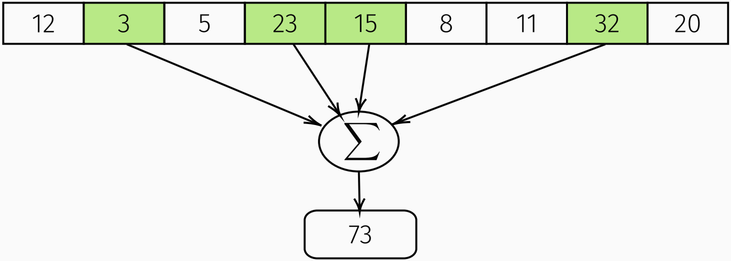
The Subset Sum Problem

Let's assume we're given a set of integers. Can we find a subset that sums up to a given target integer?

Input Set:

12	3	5	23	15	8	11	32	20

Selection of Subset:



The Subset Sum Problem

Formal definition of the problem:
Given a Multiset of integers, $S = \{x_1, x_2, x_3, \cdots, x_n\}$ and a target sum W, does there exist a subset $S' \subseteq S$ such that $\sum_{x \in S} x = W$?

DEFINITONS

Definitions

- **NP** is a complexity class that consists of decision problems (with a 'yes' answer) that can be verified by a polynomial time algorithm.
- **NP-hard** is the class of problems that are at least as hard as any problem in NP i.e. they may also be harder.
- **NP-complete** is the class of problems that are in NP and are at least as hard as any problem in NP.

Definitions

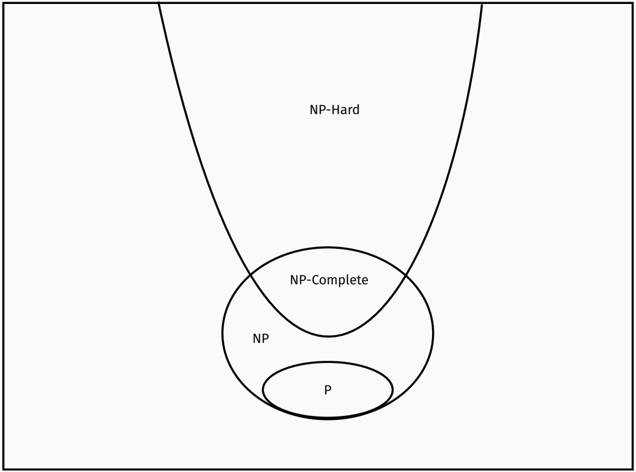


Figure: Complexity Class Diagram

OUR APPROACH

Our approach

To show that the subset sum problem is NP-complete, we will perform the following steps:

- We will show that the problem is in NP.
- We will show that the problem is NP hard.
- Therefore, we will prove that the subset sum problem lies in the intersection of NP and NP hard problems i.e. it is NP complete.

PROOF THAT THE PROBLEM IS IN NP

The subset sum problem is in NP

We first show that the subset sum problem $\in$ NP.

Proof: We take an instance of a subset sum problem with target integer W.
Let $S' = \{x_1, x_2, x_3, \cdots, x_n\}$ be a certificate of this problem i.e. the elements of S' sum up to W.
We can verify whether the sum of these integers actually sum up to W or not in polynomial time.
Therefore, the subset sum problem $\in NP$.

The subset sum problem is in NP

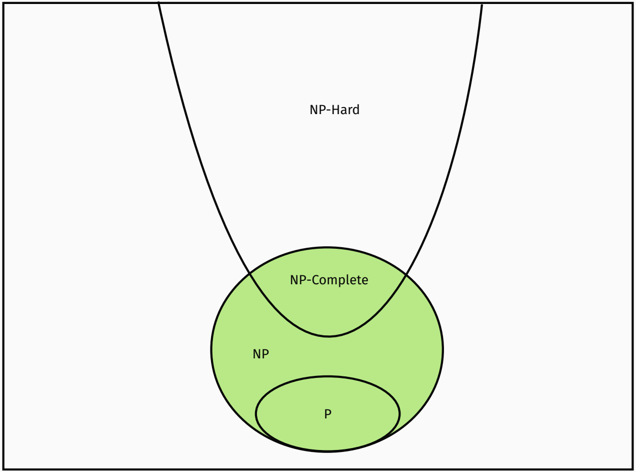


Figure: Complexity Class Diagram

PROOF THAT THE PROBLEM IS NP-HARD

Proof that the problem is NP-hard

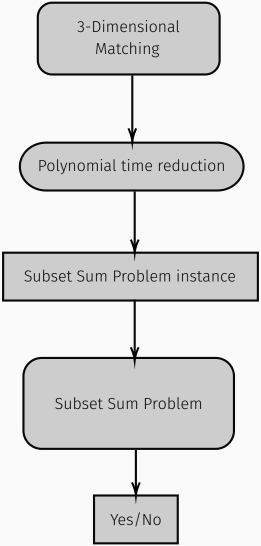
We now show that the subset sum problem is NP hard.

Our Approach: In order to show this, we will prove that the subset sum problem is as hard as any problem in NP. We will reduce a known NP-complete problem to the subset sum problem.

Logical Reasoning

- Let's assume we have a black box for solving the subset sum problem (SS).
- If we can show that an NP-complete problem (say X) can be reduced in polynomial time to SS, then it will imply that SS is at least as hard as X.
- Since the NP-complete class is a subset of NP-hard problems, so X must be a problem that is as hard or harder than any problem in NP.
- If SS is at least as hard as X, it is at least as hard as any problem in NP and hence is NP hard.

Steps of Reduction



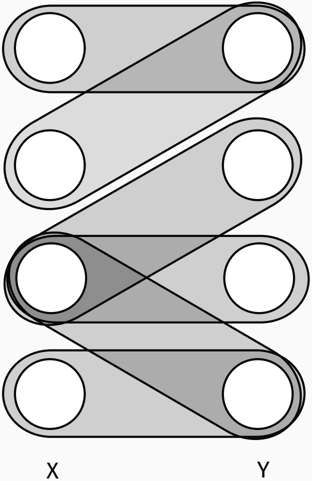
Choosing an NP complete problem

Karp in his **"Reducibility Among Combinatorial Problems"** proved the NP-completeness of 21 famous problems. They are widely known as **Karp's 21 NP-complete problems**.

One of these problems is the Three-dimensional matching problem. We will reduce it to the subset sum problem.

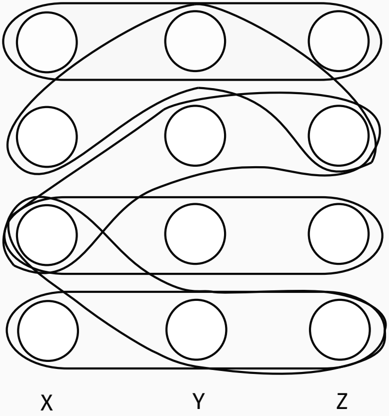
THE 3D MATCHING PROBLEM

Revisiting the 2D Matching Problem



Problem Definition: Given two sets of vertices X and Y of length n each and a set of edges or pairs E.
Does there exist a set of pairs $E' \subseteq E$ such that -

- it saturates every vertex in X and Y and
- no two edges $\in$ E' are incident on the same vertex?



Problem Definition: Given three sets of vertices X, Y and Z of length n each and a set of triples T $\subseteq X \times Y \times Z$.
Does there exist a set of triples $T' \subseteq T$ of size n such that each element is present in exactly one triple of T'?

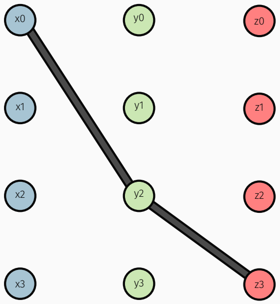
Reduction from 3D Matching to Subset Sum Problem

Reduction from 3D Matching to Subset sum problem

Given an instance X, Y, Z, T of a 3D matching, an instance of subset sum problem is constructed with set S and target W such that X, Y, Z has a 3D matching iff there is a set $S' \subseteq S$ that sums to W.

Reduction from 3D Matching to Subset sum problem

Every element in a triple can be represented as a bit vector.
Every triple $t \in T$, where $t_i = x_i, y_j, z_k$, corresponds to a number w_t of $3n$ digits. Positions $i, n+j$ and $2n+k$ receive a value of 1 whereas all the other places receive 0.



For the triple, $t_i = \{x_0, y_2, z_3\}$

x_0				y_2				z_3			
1	0	0	0	0	0	1	0	0	0	0	1

Reduction from 3D Matching to Subset sum problem

For some base $b > 1$,

$$w_t = b^i + b^{n+j} + b^{2n+k} \tag{1}$$

Reduction from 3D Matching to Subset sum problem

For some base $b > 1$,

$$w_t = b^i + b^{n+j} + b^{2n+k} \tag{1}$$

As $|T| = m$, there are at most $m\, 1's$ in a column.
To prevent addition carry the base must be greater than m.
Therefore, we set the base, $b \geq m + 1$

Reduction from 3D Matching to Subset sum problem

For some base $b > 1$,

$$w_t = b^i + b^{n+j} + b^{2n+k} \tag{1}$$

As $|T| = m$, there are at most $m\,1's$ in a column.
To prevent addition carry the base must be greater than m.
Therefore, we set the base, $b \geq m + 1$

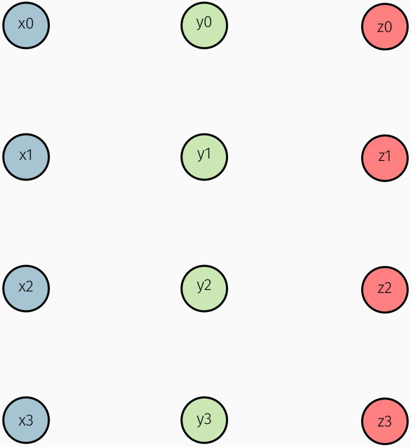
Finally, the number W defined in the subset sum problem is,

$$W = \sum_{i=0}^{3n-1} (m+1)^i \tag{2}$$

Reduction from 3D Matching to Subset sum problem

Example:

$X = \{x_0, x_1, x_2, x_3\}$
$Y = \{y_0, y_1, y_2, y_3\}$
$Z = \{z_0, z_1, z_2, z_3\}$

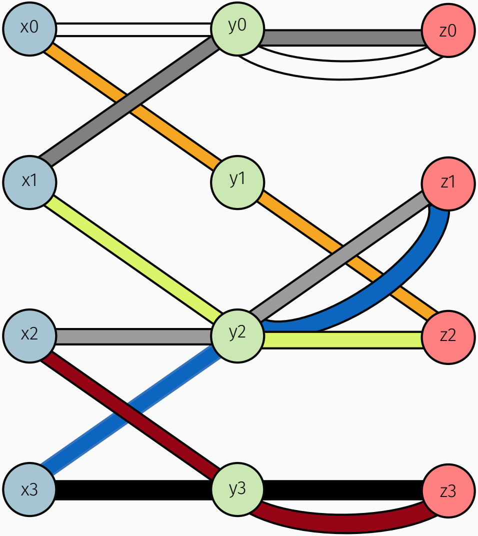


Reduction from 3D Matching to Subset sum problem

$T = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$

$t_0 = \{x_0, y_1, z_2\}$
$t_1 = \{x_1, y_0, z_0\}$
$t_2 = \{x_3, y_2, z_1\}$
$t_3 = \{x_1, y_2, z_2\}$
$t_4 = \{x_0, y_0, z_0\}$
$t_5 = \{x_2, y_2, z_1\}$
$t_6 = \{x_2, y_3, z_3\}$
$t_7 = \{x_3, y_3, z_3\}$

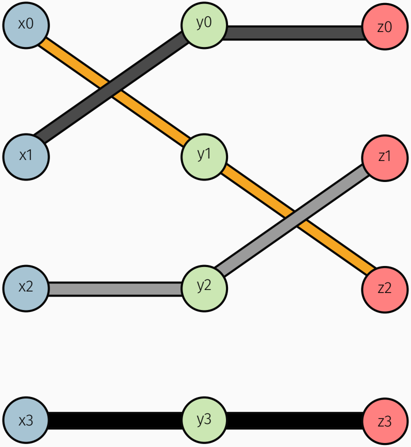
Reduction from 3D Matching to Subset sum problem



Reduction from 3D Matching to Subset sum problem

$T' \subseteq T$ is a perfect $3d$ *matching.*
$T' = \{t_0, t_1, t_5, t_7\}$

$t_0 = \{x_0, y_1, z_2\}$
$t_1 = \{x_1, y_0, z_0\}$
$t_5 = \{x_2, y_2, z_1\}$
$t_7 = \{x_3, y_3, z_3\}$



Reduction from 3D Matching to Subset sum problem

T	0	1	2	3	4	5	6	7	8	9	10	11
t_0	1	0	0	0	0	1	0	0	0	0	1	0
t_1	0	1	0	0	1	0	0	0	1	0	0	0
t_2	0	0	0	1	0	0	1	0	0	1	0	0
t_3	0	1	0	0	0	0	1	0	0	0	1	0
t_4	1	0	0	0	1	0	0	0	1	0	0	0
t_5	0	0	1	0	0	0	1	0	0	1	0	0
t_6	0	0	1	0	0	0	0	1	0	0	0	1
t_7	0	0	0	1	0	0	0	1	0	0	0	1

Reduction from 3D Matching to Subset sum problem

T	0	1	2	3	4	5	6	7	8	9	10	11
t_0	1	0	0	0	0	1	0	0	0	0	1	0
t_1	0	1	0	0	1	0	0	0	1	0	0	0
t_2	0	0	0	1	0	0	1	0	0	1	0	0
t_3	0	1	0	0	0	0	1	0	0	0	1	0
t_4	1	0	0	0	1	0	0	0	1	0	0	0
t_5	0	0	1	0	0	0	1	0	0	1	0	0
t_6	0	0	1	0	0	0	0	1	0	0	0	1
t_7	0	0	0	1	0	0	0	1	0	0	0	1
$t_0t_1t_2t_3t_4t_5t_6t_7$	1	1	1	1	1	1	1	1	1	1	1	1

The reduction is done in polynomial time

- **Input size:** 3 sets of size n each and m triples are supplied as input.
- **Construction:** For each triple, a row of size $3n$ is filled up. This is done for m triples.

Clearly, this reduction is done in polynomial time and space.

PROOF OF CORRECTNESS

Proof of Correctness

We will now show that the set of triples T contains a $3D$ matching *iff* a subset w_t of the given set S exists whose sum equals to target W.

Proof: We first assume that the set T contains a $3D$ matching.

- For every position out of $3n$ ones in the bit vector, there is exactly one addend that has 1 there.
- By construction, the sum of the numbers corresponding to the triples have 1 in each of the $3n$ digit positions.
- The sum thus equals to W.

Proof of Correctness

Conversely, we assume that the subset $w_1, w_2, ...w_p$ sums up to W. Now, $p = n$ because -

- There are exactly three 1s in each w_i.
- No carries occur.

That is, for every position out of $3n$ ones, there is exactly one addend that has 1 there. Therefore, T has a $3D$ Matching.

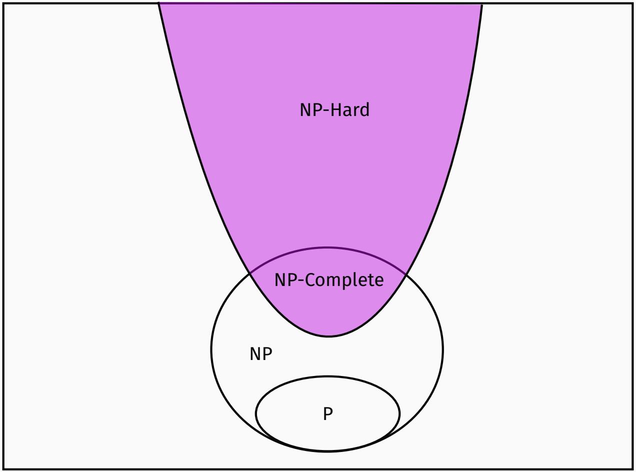


Figure: Complexity Class Diagram

Proof of NP-Completeness

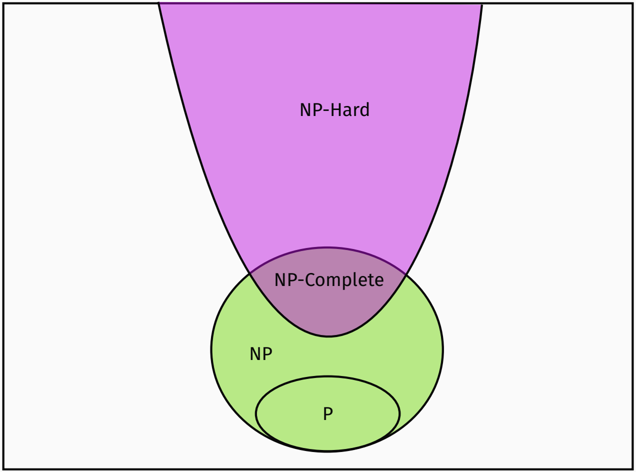


Figure: Complexity Class Diagram

REDUCTION TO OTHER PROBLEMS

Knapsack Problem

- **Instance:** A finite set of objects U, a weight $w(u) \in Z^+$, a profit $p(u) \in Z^+$ for each $u \in U$ and positive integers W (knapsack capacity) and P (desired profit).
- **Question:** Is there a subset of weights with total weight at most W, such that the corresponding profit is at least P?
- Let, $U = \{u_0, u_1, u_2, u_3, u_4\}$
  $w = \{x \in Z^+ | \ x = w(u) \ for \ u \in U\} \ = \{12, 18, 6, 21, 34\}$
  $p = \{x \in Z^+ | \ x = p(u) \ for \ u \in U\} \ = \{15, 9, 27, 9, 31\}$
  $W = 40, \ P = 50$
  $w' \subseteq w = \{12, 6, 21\}$ and $\sum_{w_i \in w'} = 12 + 6 + 21 = 39 \leq W$
  $p' \subseteq p = \{15, 27, 9\}$ and $\sum_{p_i \in p'} = 15 + 27 + 9 = 51 \geq P$

Reduction

Now we need to prove that

$$SubsetSum \leq_P Knapsack \tag{3}$$

Proof

We can easily create an instance of Knapsack problem that

$$\begin{cases} w(i) = p(i) = s(i) \\ W = P = t \end{cases} \tag{4}$$

where $s(i) \in S$ and t = target sum of a Subset Sum problem instance. The Yes/No answer to the Knapsack problem corresponds to the same answer to the Subset Sum problem. As

$$\begin{cases} \sum_{i \in S} w(i) \leq W \iff \sum_{i \in S} s(i) \leq t \\ \sum_{i \in S} p(i) \geq P \iff \sum_{i \in S} s(i) \geq t \end{cases} \iff \sum_{i \in S} s(i) = t \tag{5}$$

VARIATION OF SUBSET SUM PROBLEM

Polynomial Time Variation

SSP is solvable in polynomial time if the sequence $\{a_1, a_2, \cdots, a_n\}$ is restricted to be an arithmetic progression
The sequence can be specified concisely by the triple (a_1, n, j).

Expanded set from triple, $T = \{a, a + j, \cdots, a + (n-1)j\}$

Polynomial Time Variation

- If $S \subseteq T$ with $|S| = k$, let $SS_k = \sum_{s \in S} s$

Polynomial Time Variation

- If $S \subseteq T$ with $|S| = k$, let $SS_k = \sum_{s \in S} s$
- $SS_k = ka + mj, m \in Z^{\geq}$

Polynomial Time Variation

- If $S \subseteq T$ with $|S| = k$, let $SS_k = \sum_{s \in S} s$
- $SS_k = ka + mj, m \in Z^{\geq}$
- $c_k = MIN\{SS_k\}$ (Leftmost k elements)
  $d_k = MAX\{SS_k\}$ (Rightmost k elements)

Polynomial Time Variation

- For any k, SS_k can take any of the values from $c_k, c_k + j, c_k + 2j, \cdots, d_k - 2j, d_k - j, d_k$

Polynomial Time Variation

- For any k, SS_k can take any of the values from $c_k, c_k + j, c_k + 2j, \cdots, d_k - 2j, d_k - j, d_k$
- Increasing k causes c_k to increase

Polynomial Time Variation

- For any k, SS_k can take any of the values from $c_k, c_k + j, c_k + 2j, \cdots, d_k - 2j, d_k - j, d_k$
- Increasing k causes c_k to increase
- To achieve t from SS_k, $ka \equiv t \ (mod \ j)$ must hold

Polynomial Time Variation

- For any k, SS_k can take any of the values from $c_k, c_k + j, c_k + 2j, \cdots, d_k - 2j, d_k - j, d_k$
- Increasing k causes c_k to increase
- To achieve t from SS_k, $ka \equiv t \ (mod\ j)$ must hold
- Let k_1 be the lowest k with $d_k \geq t$

Polynomial Time Variation

- For any k, SS_k can take any of the values from $c_k, c_k + j, c_k + 2j, \cdots, d_k - 2j, d_k - j, d_k$
- Increasing k causes c_k to increase
- To achieve t from SS_k, $ka \equiv t\ (mod\ j)$ must hold
- Let k_1 be the lowest k with $d_k \geq t$
- k_1 can be found using binary search

Polynomial Time Variation

- For any k, SS_k can take any of the values from $c_k, c_k + j, c_k + 2j, \cdots, d_k - 2j, d_k - j, d_k$
- Increasing k causes c_k to increase
- To achieve t from SS_k, $ka \equiv t \ (mod \ j)$ must hold
- Let k_1 be the lowest k with $d_k \geq t$
- k_1 can be found using binary search
- Solution exists if and only if $c_{k_1} \leq t$

Other polynomial time variation

- A sub-problem of the problem Subset Sum in which $s_1, \cdots, s_k$ are the members of increasing geometric progression belongs to the class P.
- Medium-Density Subset Sum Problem belongs to polynomial time.

CONCLUSION

Conclusion

- We have discussed the class of problems: NP, NP-Hard, NP-Complete.
- We have presented problem definition of Subset Sum and 3-D Matching.
- We have proved Subset Sum problem is NP-Complete.
- We have also shown that Subset Sum can be reduced to other problems to show that they are NP-hard.
- We have shown some polynomial time variations of Subset Sum problem.
- Finally there are possible applications of subset sum such as Computer Passwords, Message Verification etc.

THANK YOU!