Given an array A with the elements a1, a2,..., an as well as a variable t, or the target sum where n is the number of elements i.e., the **SUBSETSUM** problem.

Also given is an algorithm **WILLPASS** which is an Algorithm that takes as input a set of homework with the following information.

* T array comprises t1 t2 ... tn: time taken to solve each homework (in hours)
* M array comprises m1 m2 ... mn: maximum marks of each homework (we will assume that you can be sure of getting full marks for all the homework)
* D array comprises d1 d2 ... dn: deadlines of the homework

Where n is the number of homework assignments.

* P is the passing mark.

Output YES if there is a homework schedule that lets you pass the course (which happens only when you get P or more) and output NO otherwise.

**AIM:** SUBSETSUM <= WILLPASS

**● Reduction-**

**def reduce( A, t ):**

**// Time taken to solve ith HomeWork Ti = Ai**

T = A

**// Marks to solve ith  HomeWork Mi = Ai**

M = A

**//(i.e. for all HW, time taken=marks)**

**// DeadLine for ith  HomeWork Di = t (i.e. same deadline for all HW)**

D = t

**// Passing mark equal to target t**

P = t

return T, M, D, P

**● Example-**

***Yes instance***

* A = {3, 34, 4, 12, 5, 2}, t = 9

SUBSETSUM(A, t) will return YES since the sums of A1, A3, and A6 add up to t i.e 9.

* T, M, D, P = reduce( A, t)

P = 9

| HWi | A | T | M | D |
| --- | --- | --- | --- | --- |
| 1 | 3 | 3 | 3 | 9 |
| 2 | 34 | 34 | 34 | 9 |
| 3 | 4 | 4 | 4 | 9 |
| 4 | 12 | 12 | 12 | 9 |
| 5 | 5 | 5 | 5 | 9 |
| 6 | 2 | 2 | 2 | 9 |

* WILLPASS( T, M, D, P) will return YES because there exists a sequence of homework through which a student can pass i.e HW1, HW3, HW6 which is done in total time 9 and score exactly 9 marks.
* Hence Yes instances of **SUBSETSUM** is equal to Yes instance of WILLPASS

**NO instance**

* A={3, 34, 4, 12, 5, 2}, t = 30
* No sequence exists whose sum is t, hence SUBSETSUM(A, t) will yield NO.
* T, M, D, P = reduce( A, t)

P = 30

| HWi | A | T | M | D |
| --- | --- | --- | --- | --- |
| 1 | 3 | 3 | 3 | 9 |
| 2 | 34 | 34 | 34 | 9 |
| 3 | 4 | 4 | 4 | 9 |
| 4 | 12 | 12 | 12 | 9 |
| 5 | 5 | 5 | 5 | 9 |
| 6 | 2 | 2 | 2 | 9 |

* WILLPASS( T, M, D, P) will return NO because there exists no sequence of homework through which the student may pass.
* Hence NO instances of **SUBSETSUM** is equal to NO instance of WILLPASS

**● Complexity Analysis-**

The time needed to reduce is the Time to copy n elements from array A to three n-element-arrays (T, M, and D).

That is O(n) where n is the number of elements in array A.

**● Lemma:** Given an array A & a target t, ***SUBSETSUM****( A, t) will return TRUE (i.e. there exists a subset with sum t) iff* ***WILLPASS*** *returns TRUE on the homework sets T,D,M with passing mark P. (i.e. there exists a sequence in (T, D, M, P) ←Reduce( A,t ) such that the student score at least P marks.)*

**● Proof of lemma:**

The above bi-directional statement can be broken down into two statements A and B.

So to prove the above lemma it's sufficient to prove A→B and B→A or, ~A→~B.

WILLPASS is a scheduling Algorithm that takes input as HomeWorks i.e (T, D, M, P ) and checks if there exists a sequence through which the student can score marks greater than equal to passing marks.

Let S be a sequence of HomeWorks such that marks scored by the sequence S are maximum.

Let T’ be the set of time taken by the sequence of homework S. D’ is the sequence of deadlines corresponding to sequence S, M’ is the marks obtained by sequence S, and SUM(M’) is the maximum marks obtained.

Since any homework can only be done if the time up to that homework is less than the deadline.

So SUM(T’) should be less than or equal to the maximum deadline i.e. SUM(T’)<= MAX( D’ )

Now, since Deadline D’ is the same for all homework, i.e P, implies SUM( T’ ) <=P.

Since by reduction M=T which implies M’=T’ which further implies SUM( M’ )<=P.

So for any sequence S there exist two cases:

1. SUM(M’) is equal to P.
2. SUM(M’) is less than P.

**Case a)** SUM(M’) is equal to P - which implies that there exists a sequence with the sum of marks P, Hence that student is passed so WILLPASS will return TRUE.

**Case b)** SUM(M’) is less than P - which implies that there exists no sequence with SUM(M’) equal to P. Hence that student can’t pass so WILL PASS will return False.

**Proof( A→ B)**

* if there exists a subset of A whose sum is t then according to WILLPASS (**case a)**, WILLPASS will return TRUE. [explained above]

**Proof( ~A→ ~B)**

* If there doesn't exist a subset of A whose sum is t then according to WILLPASS (**case b)**, WILLPASS will return FALSE.