H → H
G → H
C → E

R.13.1

L → M where M ∈ NPC

NO.

the fact that a problem L is Polynomial time reducible to an NP-complete problem M doesn't imply that P=PN

to prove P = NP, one would need to show that there exist a Polynomial-time algorithm that can solve any problem in NP.

R.13.13

(A) Prove that the Set Partition is NP

(1) Pick set of Integers of S and take half in S_1 and the rest in S_2.

Algorithm Set_Partition (S, S_1, S_2)

```
Sum1 := 0
for e in S1 do
    sum1 := sum1 + e.element()
sum2 := 0
for e in S2 do
    sum2 := sum2 + e.element()

if sum1 = sum2
    return YES
else
    return NO found
```

3)

1. we pass integers to w , w is a list of integers

Algorithm subSetSum (S, T, w)

    sum := 0
    for e in w do
        sum := sum + e element(s)
    if sum = T
        return true
    else
        return NO_FOUND

--> $(S, T) \rightarrow (S, S_1, S_2)$

1. {1,2,3,4,5} with target 10

- {1,2,3,4,5,15} YES WORK , but only its possible Create 1 subset sum

2.
- {1,2,3,4,5,10,15) This work

3. {1,2,3,4,5,10} NO WORK DEPENDS of the target

4. {1,2,3,4,5,10,25} Its not possible only exist 1 solve because the last element its too big

5. {1,2,3,4,5,-5} Don WORK