P1

①. If we have a solution, we just need to check whether each clause of 4CNF contains at least 1 true literal and 1 False literal.

This check only needs $4 \cdot k$ steps (k is the number of clause in 4CNF)

∴ Not-all-equal - 4CNF is a NP problem

②. We want to ~~bui~~ prove that $3SAT \le_p$ 4CNF of Not all equal.

for every clause in 3SAT, we build a clause in 4CNF by containing all literals in 3SAT and add a literal y.

$(Ex. (x_1 \vee x_2 \vee x_3))$ $(Ex. (x_1 \vee x_2 \vee x_3 \vee y))$

then we give a claim that this 3SAT satisfied iff this built 4CNF is not-all-equal.

③ $\Rightarrow$ if 3SAT is satisfied, then the literals in each 3SAT clause has at least 1 True, therefore, the 4CNF can be satisfied ∵ by choosing y=False

$\Leftarrow$ if this 4CNF is not all equal then the value of y has two choices.

1) y = False, means that the other three literals in each 4CNF clause must have at least 1 True, which means the corresponding clause in 3SAT is ~~true~~ True,

∴ 3SAT is satisfied

2) y = True, means that the other three literals in each 4CNF clause must have at ~~lease~~ least 1 False. This situation means that, the 4CNF clause also has the choice to ~~to~~ chose the negated way ~~to~~ make 4CNF of not all equal satisfied.

In other words, if y=True satisfied 4CNF of not-all-equal, it also have a choice that y=False satisfied 4CNF of NAE. Since y=False $\Rightarrow$ 3SAT is satisfied

∴ In y=True case, exist ~~some~~ literals satisfy this claim ($\Leftarrow$). q.e.d.

④ In all, $3SAT \le_p$ 4CNF of NAE, ①, 4CNF of NAE is NP ∴ 4CNF of NAE is NPC.

P2.

① This problem is in NP since we can certificate the answer one by one in R to find whether the sum of conflicts of each student is smaller ~~than~~ or equal than K

In the worst we need $K \cdot N$ steps, (search the R one by one) therefore this problem is a NP problem.

②. first we prove that 3-coloring $\le_p$ K-coloring, $K>3$

It is very simple because (1) K-coloring is a NP problem, we only need to check all the nodes in the graph and its neighboor ∴ the check time is $O(n^2)$, n is the number of graph nodes,

(2) for a graph G_1, for any 3-coloring problem, add $K-3$ nodes and connect these new nodes to all other nodes in G_1, the new grahe is G_2. (the new nodes to new nodes also need connect)

Then 1) it is easy to find that if G_2 satisfies K-coloring the new nodes must hold $K-3$ color, ∴ the other nodes satisfies 3-color ⇒ G_1 can be 3 color

2) if G_1 satisfies 3-coloring the G_2 can satisfy K-coloring by making new nodes be the other $K-3$ color.

∴ 3-coloring $\le_p$ K-coloring ∵ K-coloring is a NPC problem.

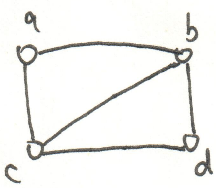
Second, we prove that K-coloring $\le_p$ this scheduling problem.

Transformation: Let the maximum number of conflict in scheduling problem be 0

Then let length of S be the K in K-coloring, elements in C be the nodes in the graph relation in R of each student be the connection of each node. Any two nodes have at most 1 connection. for example: if $R = \{\{a,b,c\},\{a,c\},\{b,c,d\}\}$

we'll add $a \Leftrightarrow b$, $a \Leftrightarrow c$, $b \Leftrightarrow c$ for 1st student, $a \Leftrightarrow c$ for 2nd student and $b \Leftrightarrow c$, $c \Leftrightarrow d$, $b \Leftrightarrow d$ for the 3rd student.

like this



(undirected graph)

③ we want to prove this scheduling problem is satisfied iff this k-coloring problem is satisfied.

⇒ if the scheduling is satisfied, means that every student does not have conflict since every edge in this undirected graph denotes a relationship of some students schedule ∴ the endpoints of each edge are not the same ∴ K-coloring is satisfied

⇐ in the same way, if K-coloring is satisfied, then the endpoints of each edge are not the same, mean the corresponding schedule is not conflict ∴ The schedule problem is satisfied

④ ∵ scheduling problem is NP and K-coloring $\le$ this problem ∴ ... is NPC.

P3.

①. if we have a solution, we only need to search from the begining of the solution to check if this loop is legal and if the ~~total~~ number of TAs is larger or equal than K. ∴ it cost at most $O(n)$ times n is the number of all TAs. ∴ this problem is a NP problem

②. we want to prove: DIR-HAM-CYCLE $\leq_p$ TA cycles with at least K TAs.
build an instance of TA cycles, we let the number of ~~tot~~ all TAs are K,
and transform it to a directed-graph, where each TA is the node, the connection of this graph depends on the relationship of TAs, if A_i work as a TA for A_j then we create $A_i \to A_j$ in this graph.

③ we want to prove: the created graph (DIR-HAM-CYCLE) exists a simple directed cycle that contains all the k TAs iff a simple TAs cycle contains at least K TAs

⇒ if the graph has a simple cycle contains all TAs, we can follow the directed path (or search all the nodes) one by one to show that this is a TA cycle contains K TAs, since in the simple cycle, every node only appears once ∴ it also be a simple TA cycle

⇐ similarly, if a simple TAs cycle contains at least K TAs, ~~in this~~
∵ all TA appears once, ∴ we can follow the path in the graph and create a simple directed cycle.

④ In all, this problem is a NP problem and DIR-HAM-CYCLE $\leq_p$ this problem
∴ this problem is NPC.

P4.

① If we have the solution, we just need to check whether the solution can be found in the ~~objects~~ items, and to check whether the selected ~~goods~~ items have total weight $\le b$ and total profit $\ge k$. This check need at most $O(n^2)$ steps $\therefore$ This problem is a NP problem

② We want to show Subset Sum $\le_p$ Knapsack.

We build an instance of knapsack that: • contains n items

and $\begin{cases} a_i = c_i = s_i \\ b = k = t \end{cases}$ [Here s_i is the value of elements in subset sum

t is the goal of subset sum.

③ We give a claim that Subset sum satisfied iff this knapsack satisfied.

for the knapsack, we have $\begin{cases} \sum_{i \in S} a_i \le b \\ \sum_{i \in S} c_i \ge k \end{cases} \Longleftrightarrow \begin{cases} \sum_{i \in S} s_i \le t \\ \sum_{i \in S} s_i \ge t \end{cases} \Longleftrightarrow \sum_{i \in S} s_i = t$

Here S is the set of selected items

$\therefore$ $\Rightarrow$ If there exist a set S satisfied the subset-sum,

then this set also satisfies $\begin{cases} \sum_{i \in S} a_i \le b \\ \sum_{i \in S} c_i \ge k \end{cases}$ $\therefore$ this knapsack is satisfied.

$\Leftarrow$ If there exis a set S satisfies the knapsack.,

then ~~there exis~~ this set also satisfies $\begin{cases} \sum_{i \in S} s_i \le t \\ \sum_{i \in S} s_i \ge t \end{cases} \Longleftrightarrow \sum_{i \in S} s_i = t$

$\therefore$ the subsetsum is satisfied

④ In all Subset sum $\le_p$ knapsack, knapsack is a NP problem

$\therefore$ knap~~set~~sack is NPC

P5. Note, the prve of 4CNF ~~a~~ of not all equal is shown in the next page

① If we have a other scheme that satisfies this requirement it can be easily checked by checking each students, judge whether they have at least 1 red and 1 blue balloon. In this worst case, we need $O(k\cdot m)$ time, k is the maximum ~~number~~ number of ~~this~~ length(S_i) for all $i \in m$.

∴ The BCP (Balloon colering problem) is a NP problem

②. We want to prove that 4CNF of NAE $\leq_p$ BCP

~~Suppose~~ We generate an Instance of BCP:

Let $|S_i| = 4$ for all $i \in n$ (all ~~stude~~ children)

So the BCP p is: we have m children, each children ~~see~~ selected different 4 ballons denoted by $S_1, S_2 \cdots S_m$, $S_i \in F$, does this selection makes each children have at least 1 blue and 1 red balloon.

Transform red, blue to True, False in 4CNF, the selation of each child is the clause of 4CNF. Then,

③ We want to prove: this BCP is satisfied iff this 4CNF of NAE satisfied.

⇒ if this BCP satisfied, means that each child (clause) has at least 1 red (True) and 1 blue (False) ∴ The 4CNF of NAE satisfied

⇐ if 4CNF of NAE satisfied, means that each clause (children) has at least 1 True (red) and 1 False (blue) ∴ The BCP is satisfied.

④ In all, BCP is a NP problem and ~~BCP~~ $\underbrace{3SAT \leq_p 4CNF \text{ of } NAE}_{\text{by Problem 1 in next page}} \leq_p BCP$

∴ BCP is NPC

P6.

① This problem can be easily certificated because we only need to follow the path and check whether this path is legal and whether the distance is exactly l, so it can be certificated in polynomial time ($O(n)$) n is the number of bridges, this problem is a NP problem

② We want to prove that subset sum $\le_p$ RP (RP means this robot path problem)

Here we build a RP problem with only 1 room (v), and this room has n self-loop bridges (n is the number of elements in subset sum)

and the length of each bridges is equal to the value of corresponding integer value

(i.e., $d_i = v_i$, d_i is the length of ith bridge, v_i is the value of i^{th} element)

the goal in subset sum, g, is equal to the travelling distance l.

③ we want to prove: this RP problem is satisfied iff subset sum is satisfied.

$\Rightarrow$ if there exists a set S that $\sum_{i \in S} d_i = l$, $\because d_i = v_i$, $l = g$

$\therefore \sum_{i \in S} v_i = g$ $\therefore$ subset sum is satisfied.

$\Rightarrow$ if there exists a set S that $\sum_{i \in S} v_i = g$ $\because d_i = v_i$, $l = g$

$\therefore \sum_{i \in S} d_i = l$ $\therefore$ there exist a path that ~~satisfy~~ satisfies RP problem

$\therefore$ this claim is true

④

In all we have subset sum $\le_p$ Rp, Rp is a NP problem

$\therefore$ Rp is NPC.