# CSC341 HW11

April 23, 2024

**Havin Lim**

## Academic Honesty

### Written Sources Used:

Michael Sipser - Introduction to the Theory of Computation

### Help Obtained:

None

## Question 1

$h$ would be $= g(f(x))$, where $x \in A$. The TM $M_h$ will be a TM that computes $f(x)$ and uses the result of $f(x)$ on to $M_g$.

To show that the mapping reduction is correct, if we assume that $x \in A$, this means that $f(x) \in B$, which leads to $g(f(x)) \in C$. Overall, this makes $h(x) \in C$.

$M_h$ runs in polynomial time because it is a combination of two polynomial time computations. Since $f$ and $g$ both run in polynomial time, this guarantees that $M_h$ would run in polynomial time as well.

## Quesiton 2

### (1)

Certificate: Assigning a single color for each vertex in the graph $G$.
Verification: Check that no two adjacent vertices in graph $G$ have the same color. In this verification process, checking each edge will happen in constant time, but the number of edges is polynomial regarding the number of vertices which makes it polynomial time.

### (2)

Certificate: Schedule that assigns each job $J_i$ to a processor with start and end times.
Verification: Check for every pair $J_m \prec J_n$, the end time of $J_m(e_m) \leq$ the start time of $J_n(s_n)$. In addition, make sure that there are no overlapping two jobs on the same processor, and make sure that the end time of the last job on each of the processors doesn't exceed $t$. The process of sorting jobs per processor and checking if there are any overlaps happens in polynomial time and checking if it exceeds $t$ or not is linear, making the overall verification process polynomial time.

## Question 3

For KNAPSACK to be NP-complete we need to prove that KNAPSACK is in NP and that subsetsum is polynomial-time reducible to KNAPSACK.

(1) KNAPSACK is in NP
Certificate: $B' \subseteq B$
Both $\sum_{b \in B'} w(b) \leq W$ and $\sum_{b \in B'} v(b) \leq V$ requires adding up the elements in $B'$ which takes polynomial time.
(2) subsetsum can be reduced in polynomial time to KNAPSACK
We define a KNAPSACK instance with $B = S$, $W, V = t$, and for each $s \in S$, $w(s), v(s) = s$.

If there exists a subset $S' \subseteq S$ where $S' = t$, then KNAPSACK conditions are also met. Mapping each element of $S$ and assigning values to $W$ and $V$ takes linear time relatvie to the number of elemetns in $S$, therefore it takes polynomial time.