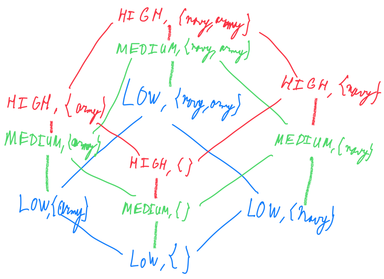
Homework 3: Mandatory Access Control

Principles of data protection

Lattice of classifications

Note: we have the Biba model, so this is about integrity.
Additionally, we have low-watermark for subjects;
this means that subjects *can read down*, but doing so **drops their integrity level**.



Rules: 1. Subject can write object iff it dominates the object $\lambda(s) \geq \lambda(o)$
2. Subject can read all objects
3. after read, we get $\lambda(s) = glb(\lambda(s), \lambda(o))$

Questions 1.2 & 1.3

2. Yes, the colonel can change the number of navy units, since his integrity class dominates that of the number of navy units: (HIGH, {Navy}) ≥ (MEDIUM, {Navy}). ~~However, this would decrease the colonel's integrity level to (MEDIUM, {Navy}).~~ Warning: ***this red-coloured part is not correct. Only a read decreases the integrity level!***

3. No, the colonel cannot change the number of navy units after reading the cost of navy units. Note that the cost of navy units has integrity class (LOW, {Navy}), which means that, after reading this object, the colonel's integrity class decreases to (LOW, {Navy}). This makes them unable to write to the number of navy units, which has integrity class (MEDIUM, {Navy}), and we have (MEDIUM, {Navy}) ≥ (LOW, {Navy}).

NEEDS FIXING

Questions 1.6 & 1.7

6. Yes, the captain can compute the cost of overall defence units. Note that reading information does not require belonging to any integrity class in the Biba model *with low-watermark for subjects*, and hence, the necessary information can be read.

7. This question is problematic; while the soldier can read the number of navy units at any time, he cannot write to/modify it, since he only has the LOW integrity level, while writing to the number of navy units requires the MEDIUM integrity level. Hence, reading the number of navy units after modifying it is not possible *(since the modification fails)*.

Question 2

- With n security levels and m compartments, we can construct $n \cdot 2^m$ security classes.
  - With m compartments, one can construct a (sub)lattice consisting of 2^m classes, since, for every component, a given subject/object is either in that compartment or not, and we can have any combination. This gives 2^m classes due to being equivalent to counting how many bit strings of length m can be made.
  - We then multiply this by n levels, since these sub-lattices can essentially be duplicated for every level/put above each other, where the resulting ordering is similar to that in the solution to question 1.1.