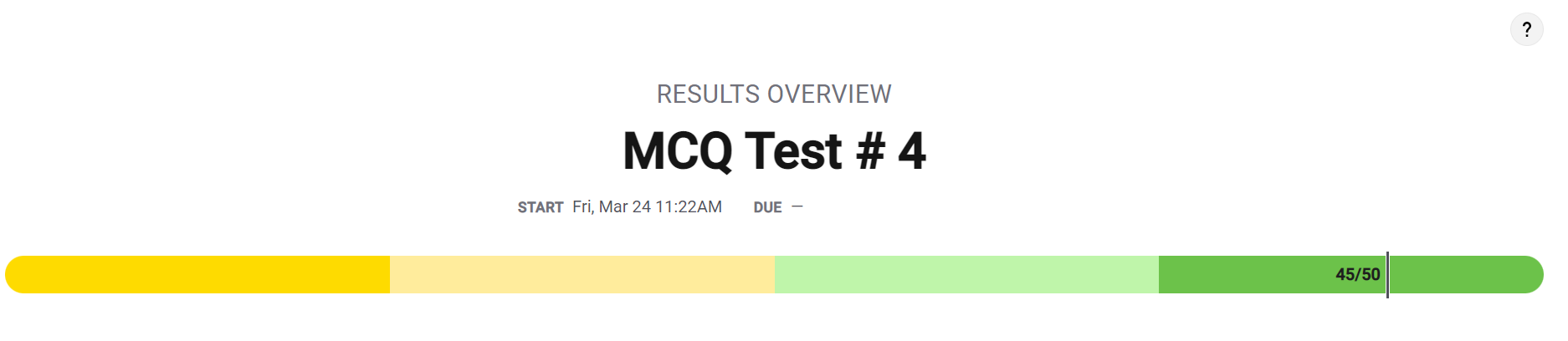
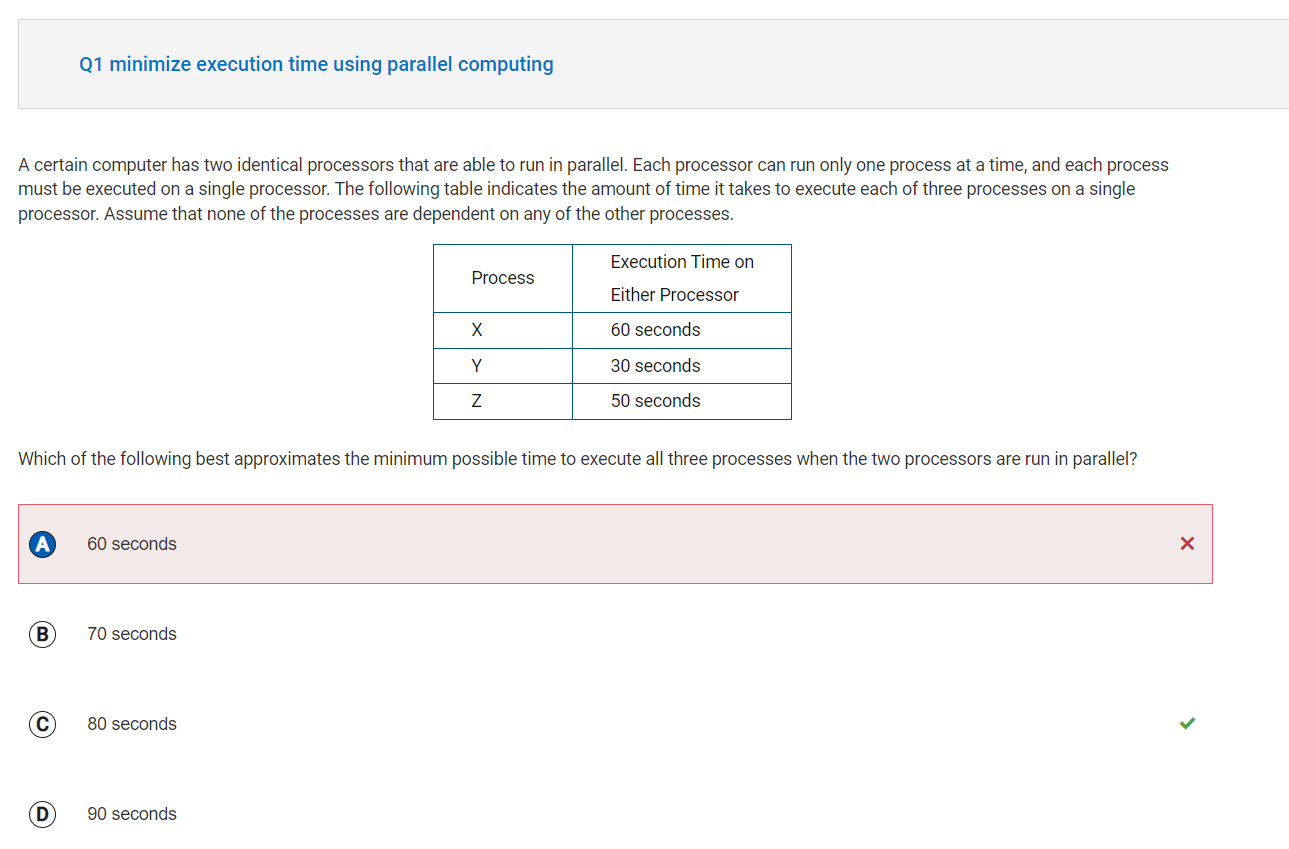
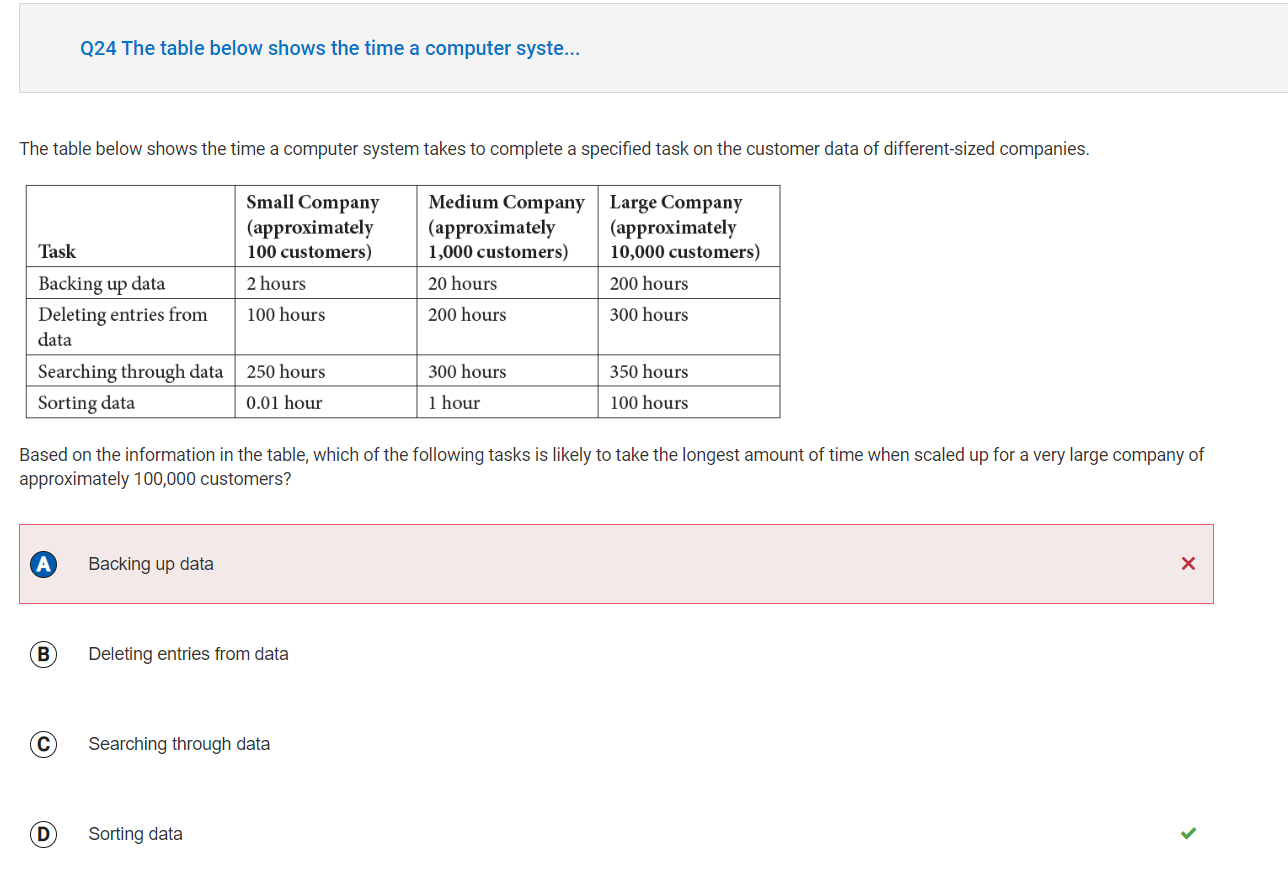
Shreya Sapkal

Period 5

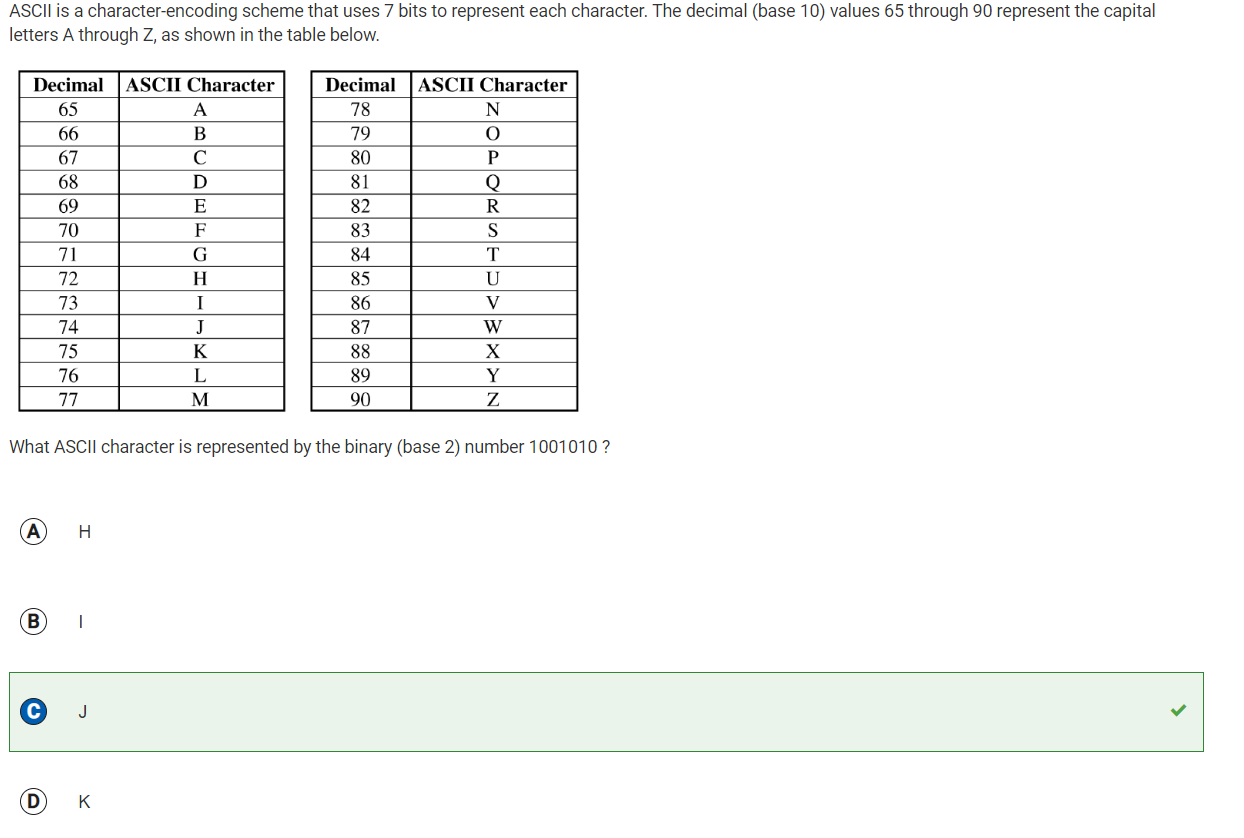




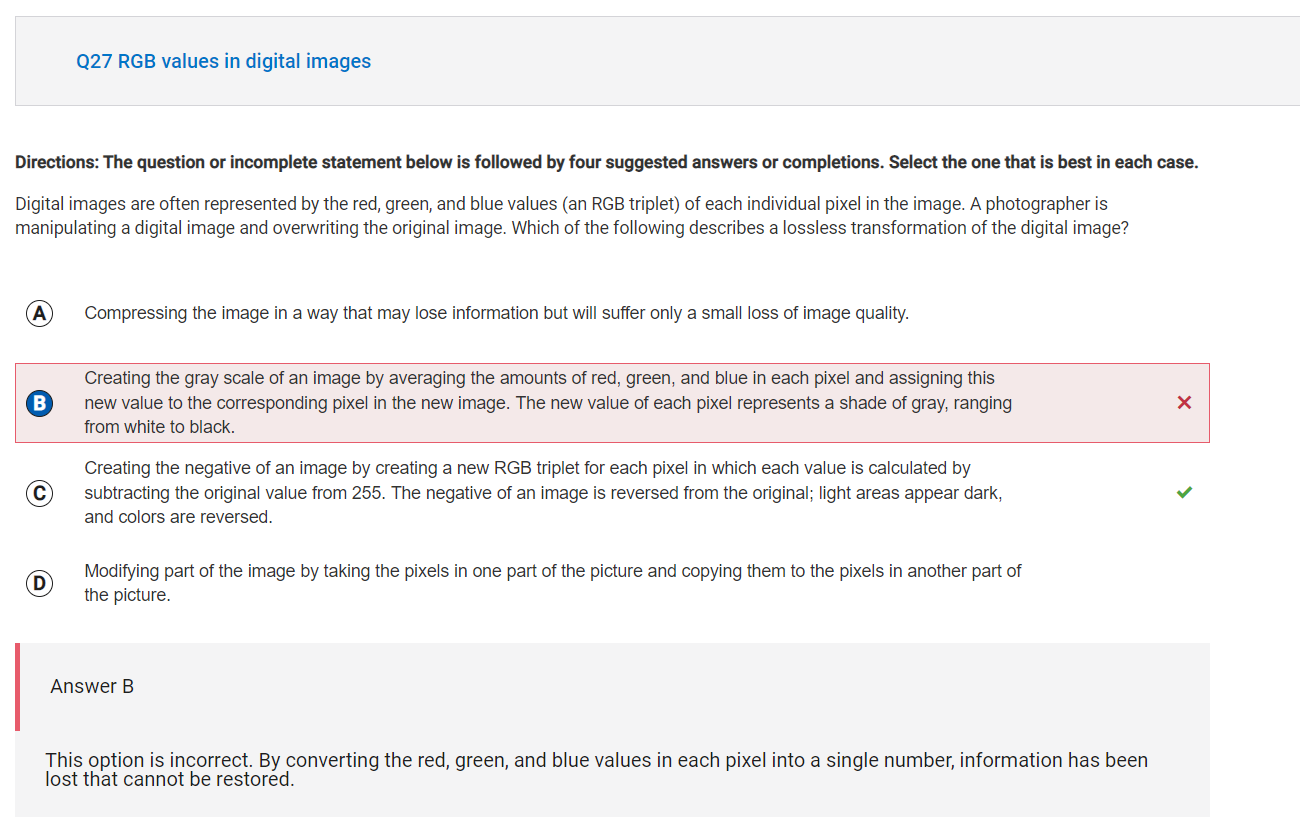
Reflection: I am not entirely sure why I got this problem wrong. I think I would benefit from some clarification.

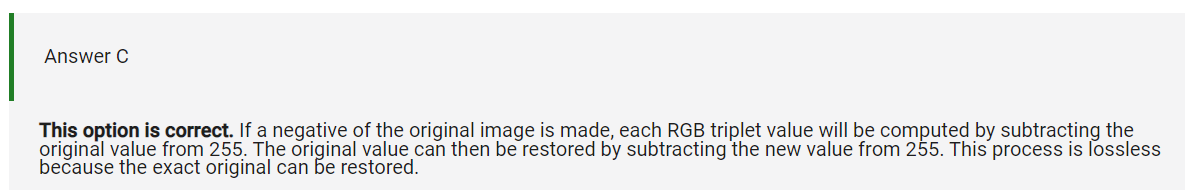


Reflection: I selected “backing up data” because it was increasing exponentially as company size increased. However, “sorting data” is getting multiplied by 100 each time the company size goes up by 10 times, and therefore increases in time the most, if considering proportionality to increase in company size.

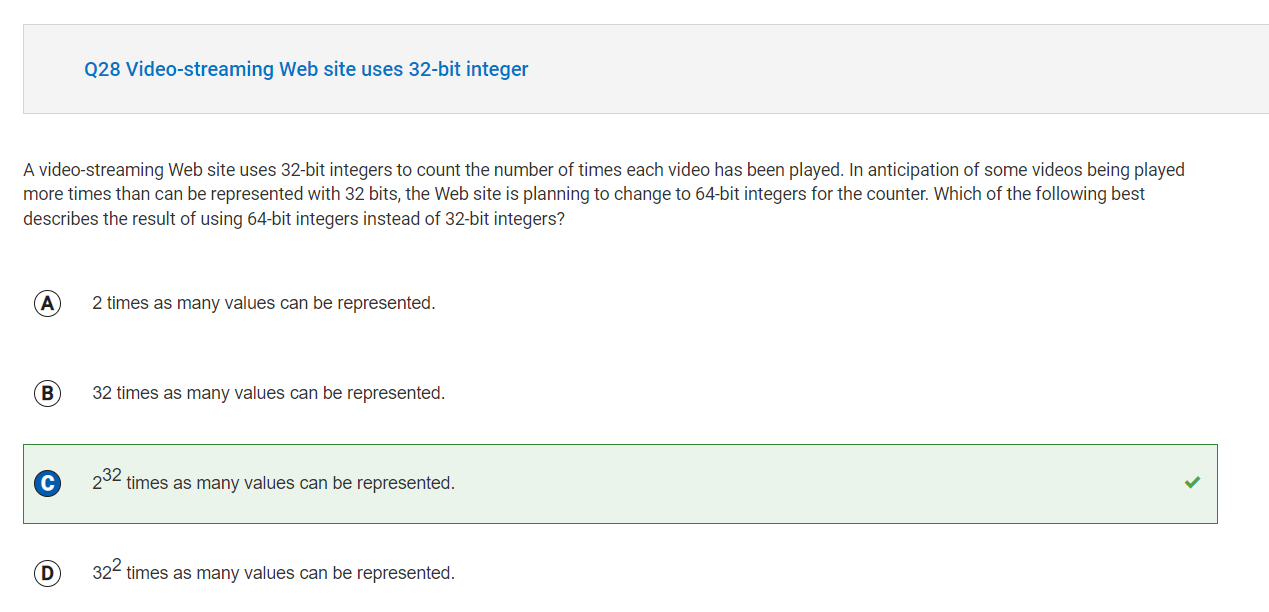


Reflection: I decided to blog this question because it was confusing to me at first. I learned that the binary (base 2) number 1001010 is represented by the decimal value 74, which on the ASCII table is represented by the letter J.

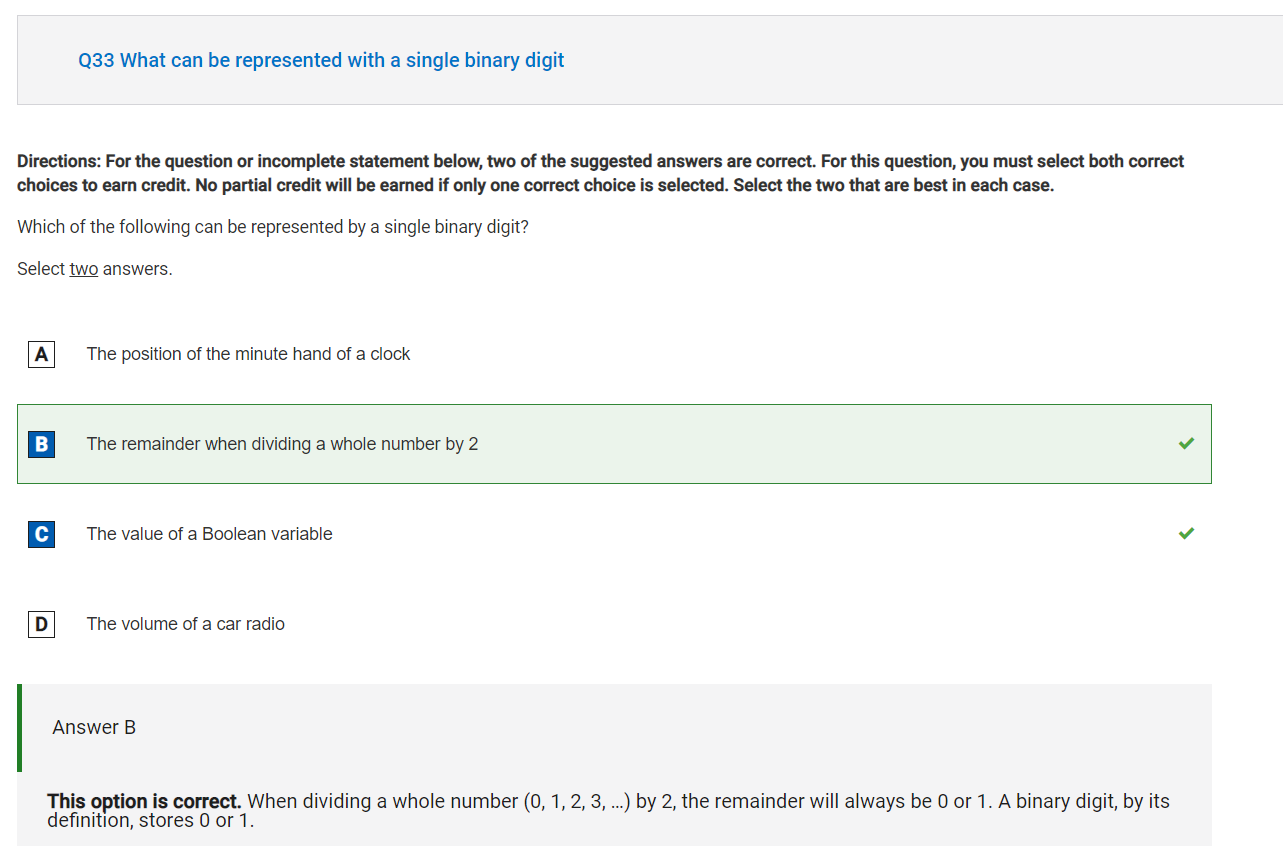




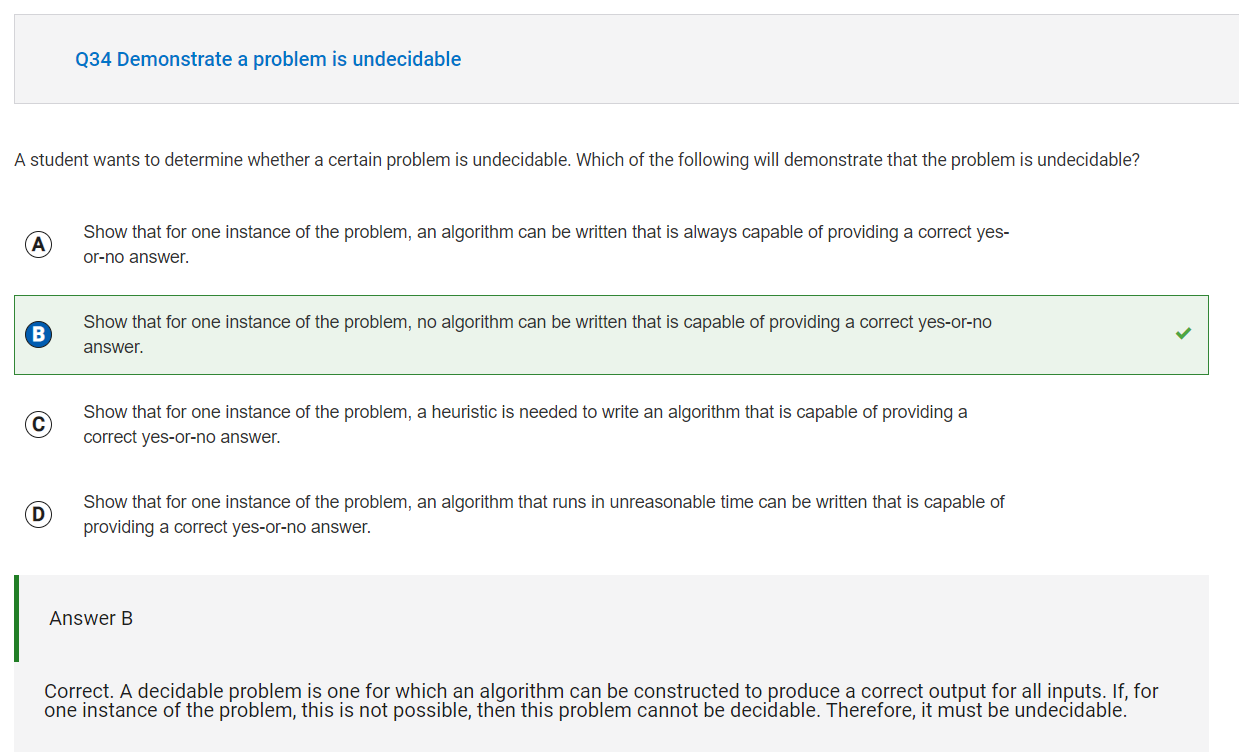
Reflection: I didn’t realize that this problem was essentially a lossy vs. lossless problem. I was supposed to look for the answer that would result in the least amount of data being lost with the alteration of the image. Subtracting from the original value can be reversed, and the exact original can be restored. However, with option B, the original image will be altered and in order to revert it, we would have to create a new image.



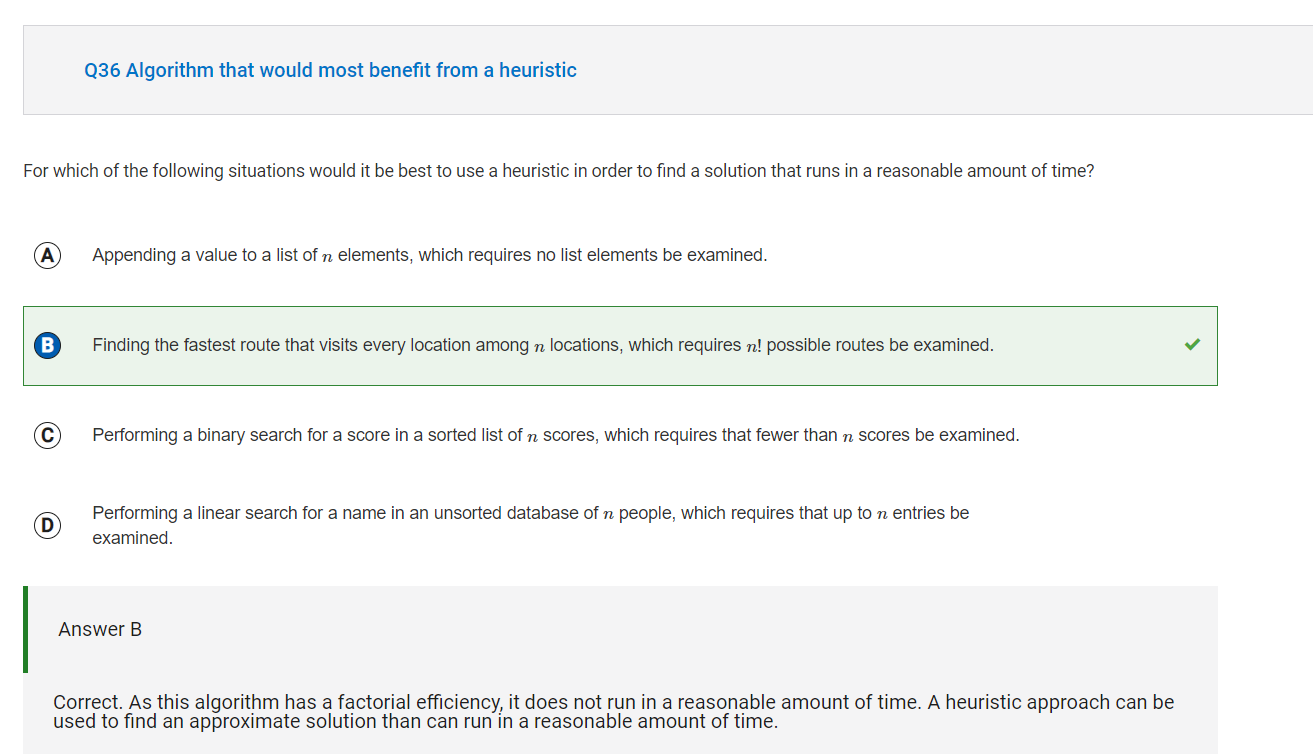
Reflection: I blogged this problem because I need more work on representation of data through bits.



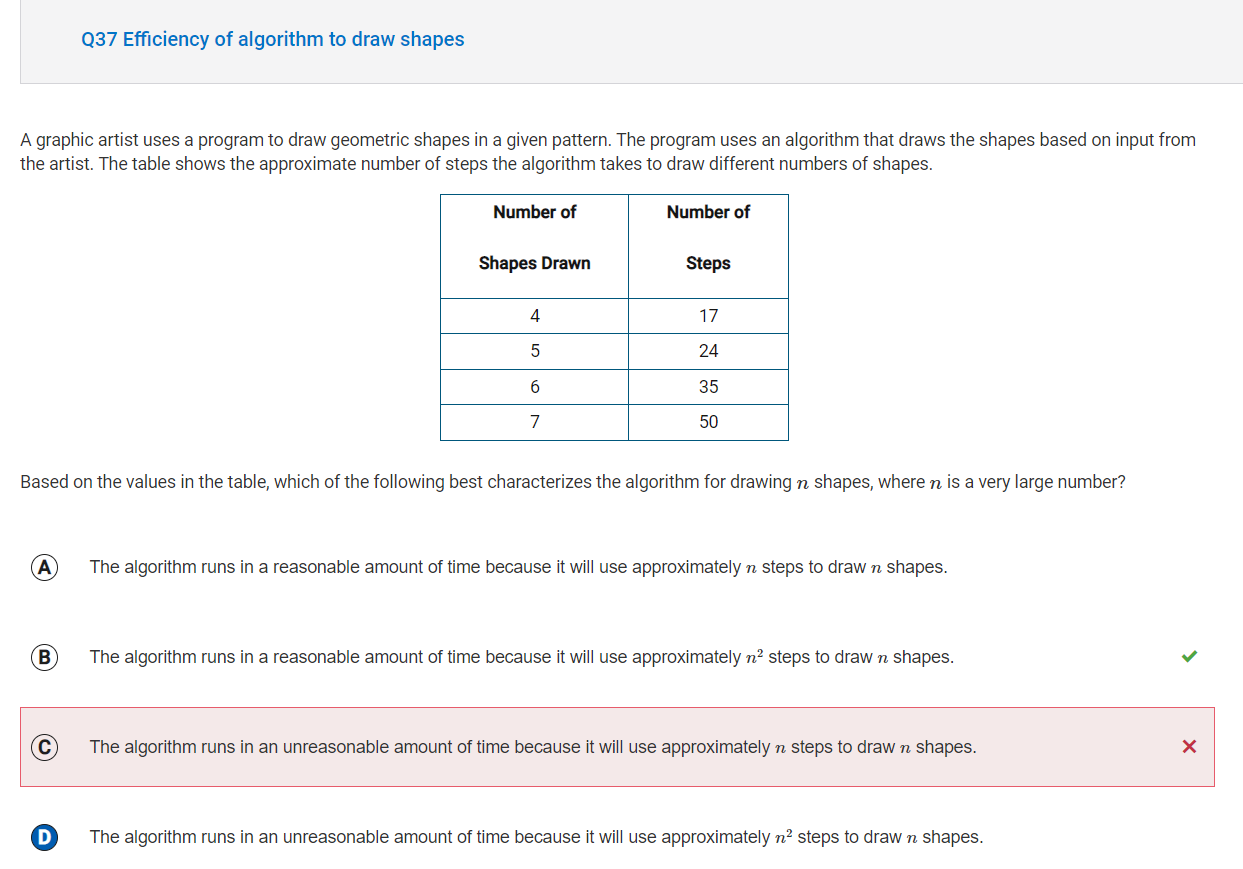
Reflection: I blogged this question, because it reminded me that a single binary digit is essentially a 1 or a 0. Now I have a stronger understanding of how I got this problem correct. The value of a Boolean variable can be stored as a single binary digit, because “true” could be 0 and “false” could be 1, or vice-versa.

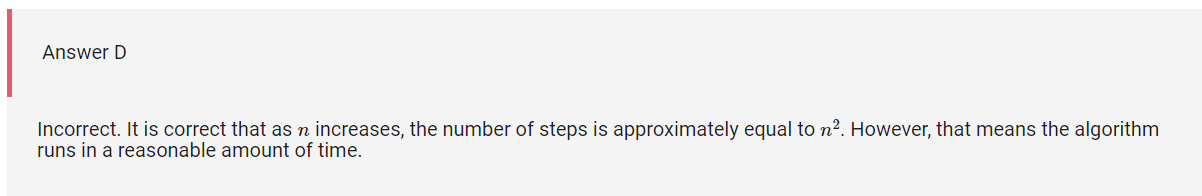


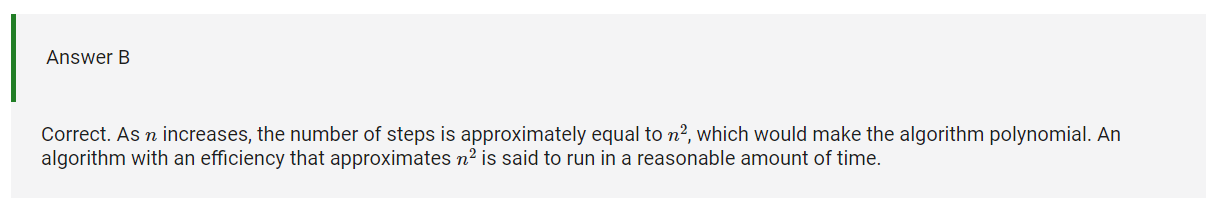
Reflection: From this problem, I learned that a decideable problem is one for which an algorithm can be constructed to produce a correct output for all inputs. If even for one input there is an incorrect output, then the problem is undecideable. In order to prove whether or not a problem is decideable, we must show that for one instance of the problem, there is no algorithm that can be written that would provide a correct answer. An undecidable problem is one for which no algorithm can be constructed that is always capable of providing a correct yes-or-no answer.



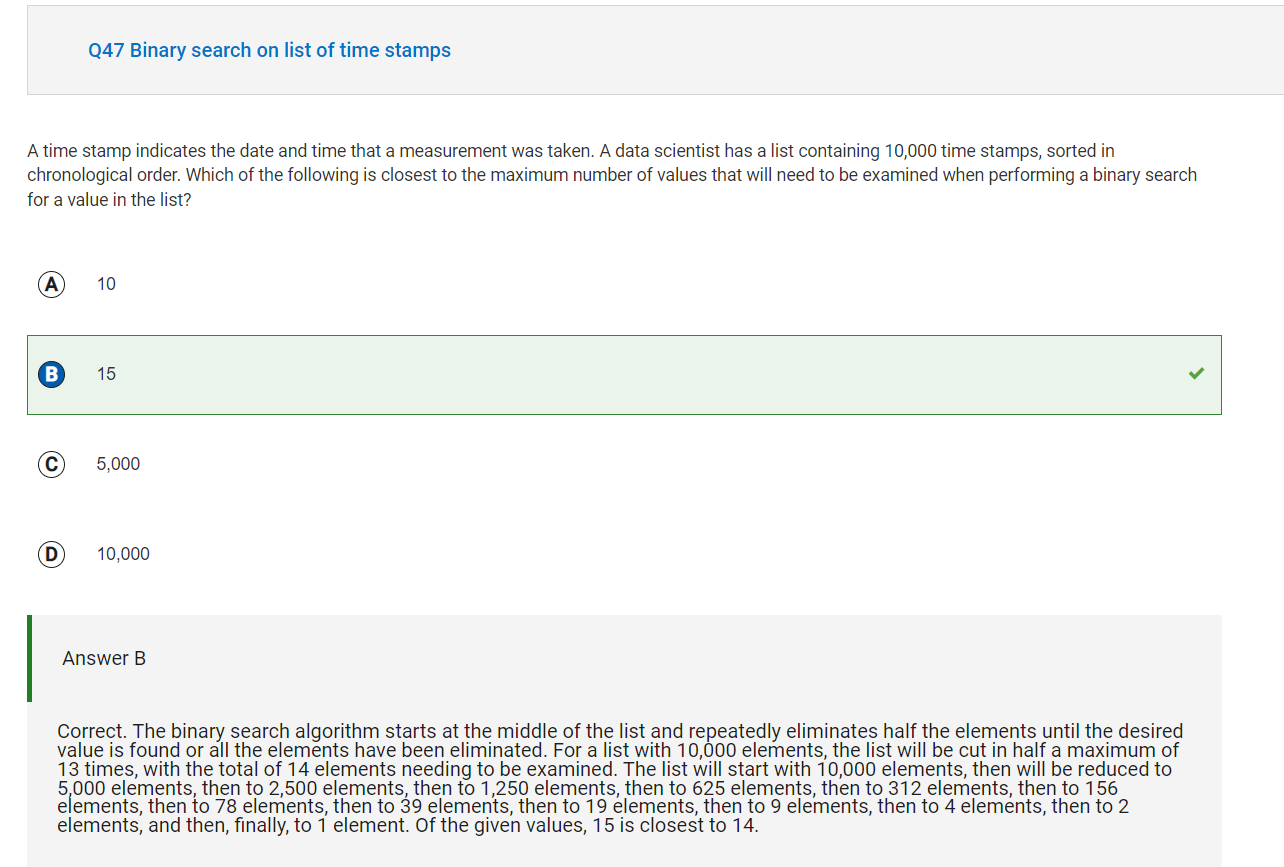
Reflection: I chose to blog this problem because from it, I learned that an algorithm with factorial efficiency doesn’t run in a reasonable amount of time.



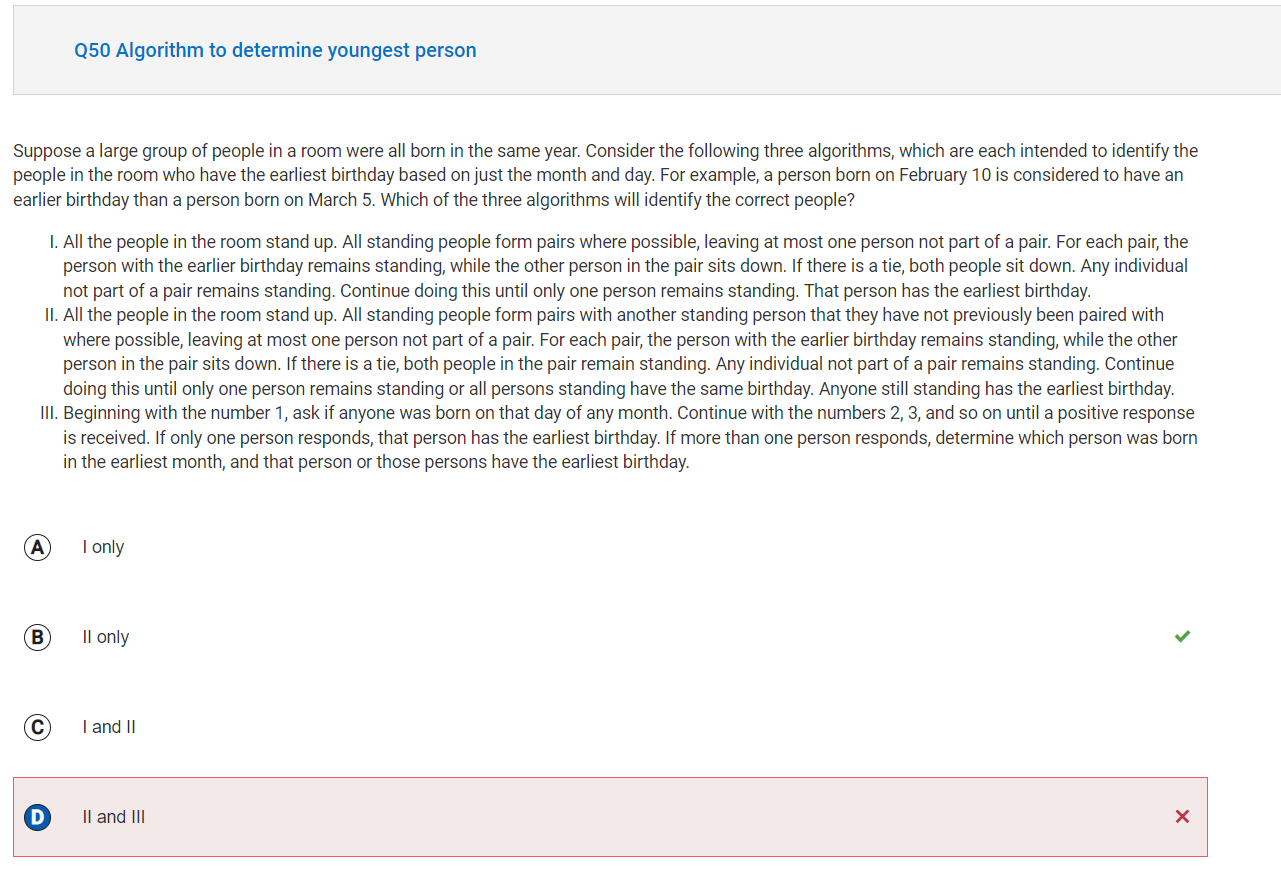


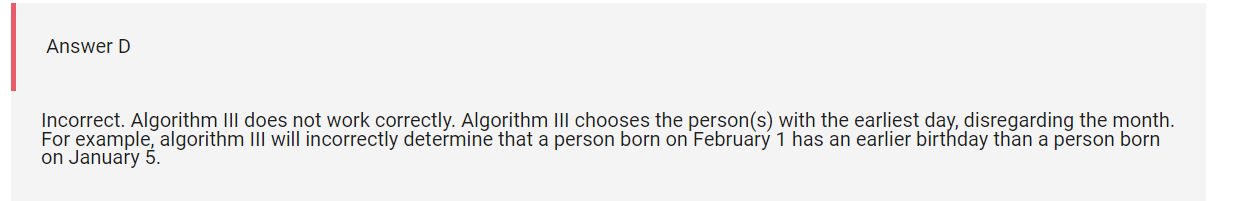


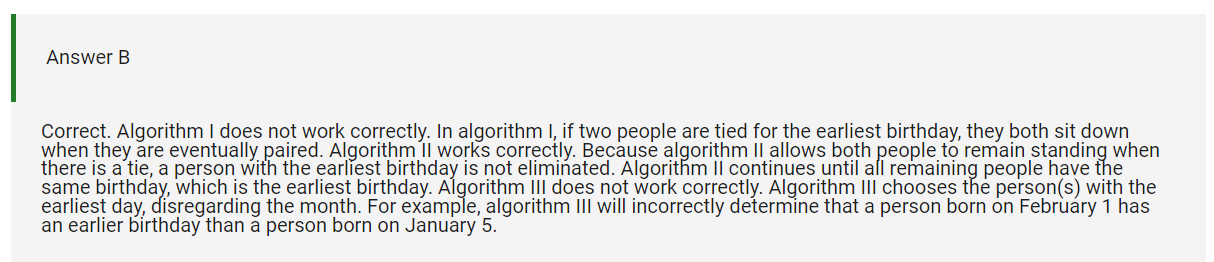
Reflection: For this problem, I knew that as n increases, the number of steps increases by about n^2. I learned that this means that the algorithm is polynomial, and an algorithm that approximates an efficiency of n^2 is supposed to run in a reasonable amount of time.



Reflection: Blogged this question because it was confusing.







Reflection: I knew that algorithm 1 didn’t work properly, but I thought that 3 did as well as 2.