CSCE 222: Homework 02

**29.**

**a) Use pseudocode to specify a brute-force algorithm that**

**determines when given as input a sequence of n positive**

**integers whether there are two distinct terms of**

**the sequence that have as sum a third term. The algorithm**

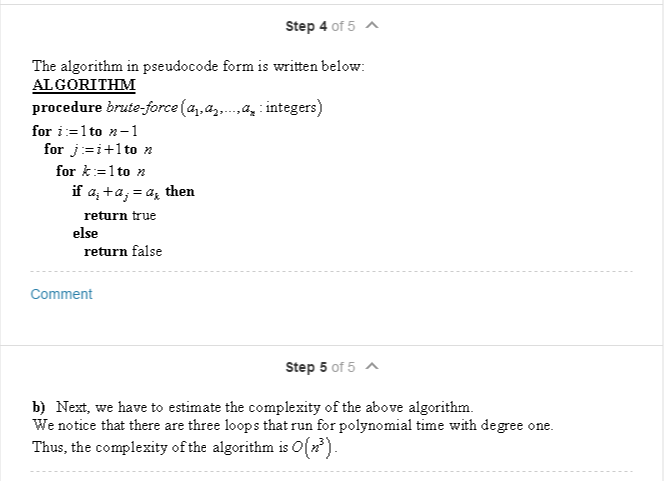
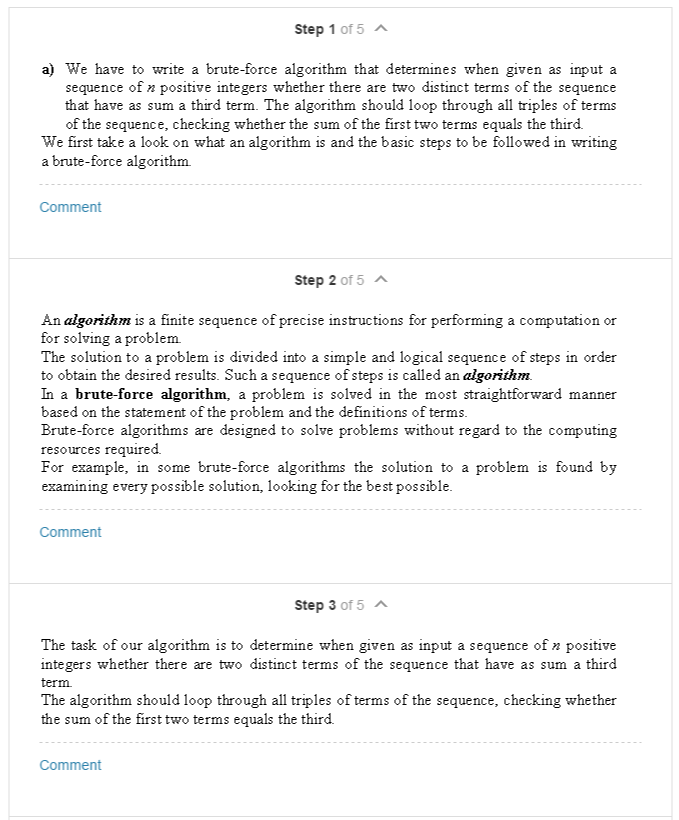
**should loop through all triples of terms of the**

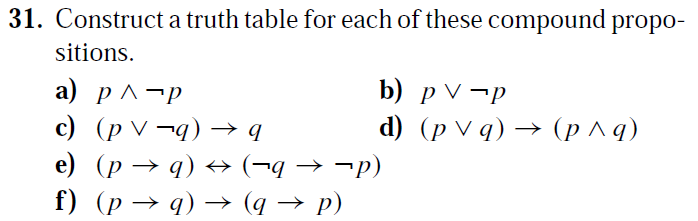
**sequence, checking whether the sum of the first two**

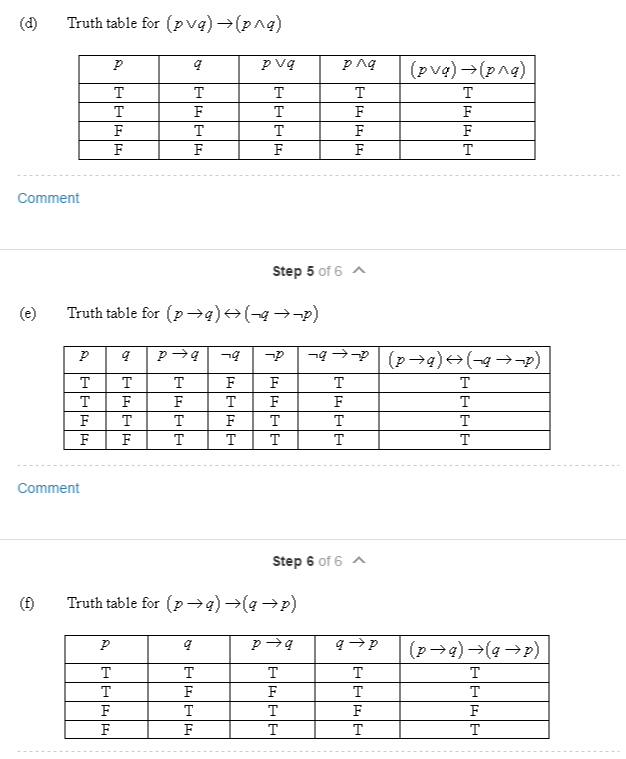
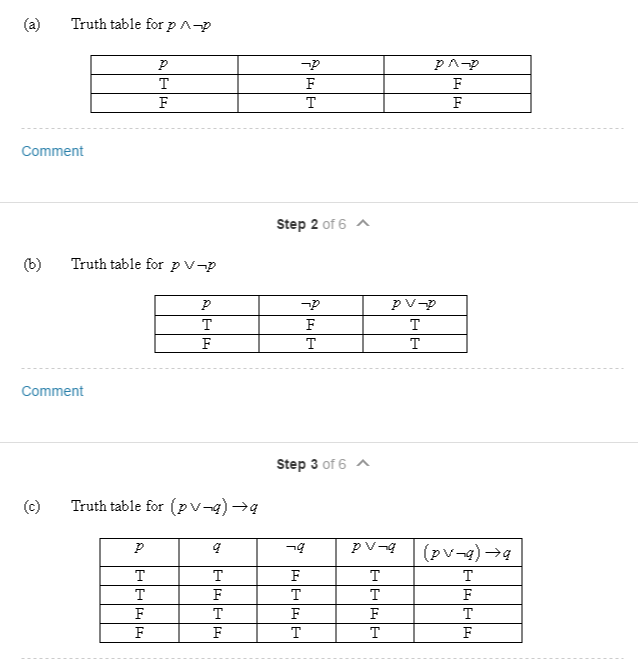
**terms equals the third.**

**b) Give a big-O estimate for the complexity of the bruteforce**

**algorithm from part (a).**







Exercises 19–23 relate to inhabitants of the island of knights

and knaves created by Smullyan, where knights always tell

the truth and knaves always lie. You encounter two people,

*A* and *B*. Determine, if possible, what *A* and *B* are if they

address you in the ways described. If you cannot determine

what these two people are, can you draw any conclusions?

**Exercises 19–23 relate to inhabitants of the island of knights**

**and knaves created by Smullyan, where knights always tell**

**the truth and knaves always lie. You encounter two people,**

**A and B. Determine, if possible, what A and B are if they**

**address you in the ways described. If you cannot determine**

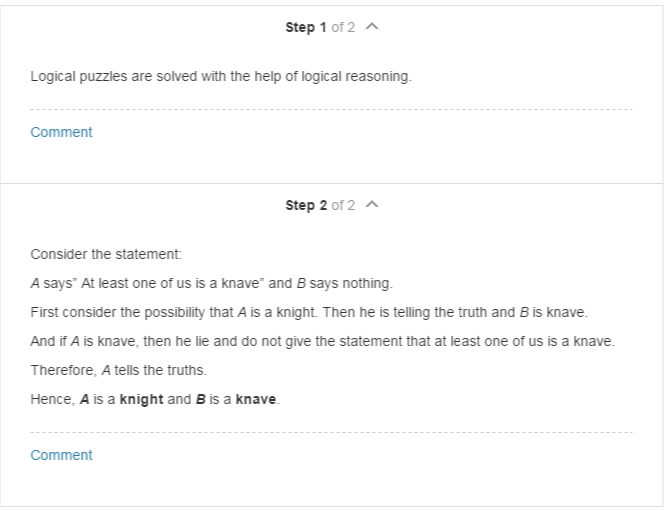
**what these two people are, can you draw any conclusions?**

**19. A says “At least one of us is a knave” and B says nothing.**

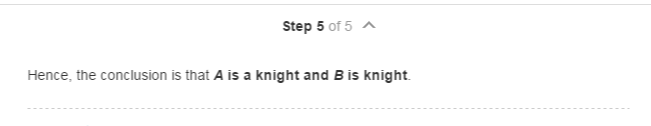
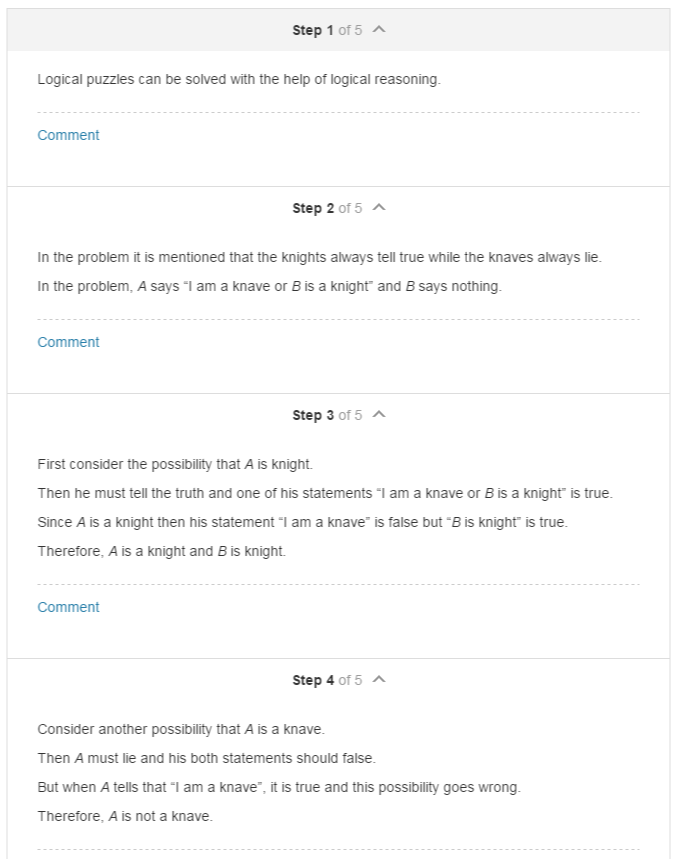
**21. Asays “I am a knave or B is a knight” andB says nothing.**

**23. A says “We are both knaves” and B says nothing.**

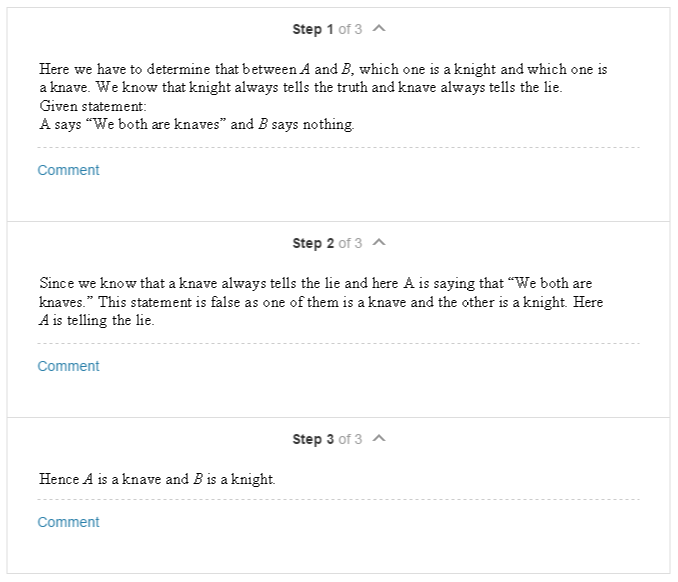
**19**



**21**



**23**



**50.** In this exercise we will show that {↓} is a functionally

complete collection of logical operators.

**a)** Show that *p* ↓ *p* is logically equivalent to ￢*p*.

**b)** Show that *(p* ↓ *q)* ↓ *(p* ↓ *q)* is logically equivalent

to *p* ∨ *q*.

**c)** Conclude from parts (a) and (b), and Exercise 49, that

{↓} is a functionally complete collection of logical

operators.

**∗51.** Find a compound proposition logically equivalent to

*p* → *q* using only the logical operator ↓.