This document is to provide how you would answer some questions from lecture if they were homework or exam questions instead. The text below, in black, would be sufficient for full credit if the Edit Distance problem, presented in lecture, were instead a homework question.

Define Edit$(i, j)$ to be the minimum cost to convert $X[1 \ldots i]$ to $Y[1 \ldots j]$.

```
Edit(i, j)
  if i == 0 then
    return j
  else if j == 0 then
    return i
  ins = 1+ Edit(i, j − 1)
  del = 1+ Edit(i − 1, j)
  if X[i] == Y[j] then
    sub = Edit(i − 1, j − 1)
  else
    sub = 1+ Edit(i − 1, j − 1)
  return min(ins, del, sub)
```

This can be stored in a two-dimensional vector, Edit$[0 \ldots n, 0 \ldots m]$, where $n$ and $m$ are the lengths of the two input strings $X$ and $Y$, respectively. This can then be filled in first by the base cases, and then the general cases can be filled in by increasing value of the first parameter, then increasing value of the second. Each recursive case takes $\mathcal{O}(1)$ to fill in, and there are $\mathcal{O}(nm)$ cases, for a total of $\mathcal{O}(mn)$ time.

3-SAT($n$ variables, $k$ clauses)
  **for** each clause $A \vee B \vee C$ **do**
    Create 3 vertices // one each $A, B, C$
    Add edges $(A, B), (B, C), (A, C)$
  **for** each variable $x_i$ **do**
    Connect any "$x_i$" node to all $\overline{x_i}$ nodes
  **return** INDEPENDENT SET($G, k$)

*Explanation:* The idea is that each clause forms a clique, from which at most one vertex can be chosen by the Independent Set solver. Because there are $k$ clauses and $k$ cliques, at most one in each will be chosen. The chosen vertex represents a literal to set to be true or false to satisfy the clause. We add edges between literals and their negations so prohibit them from both being chosen.

*No false positives*: a return value of true indicates that there are $k$ vertices that are chosen. Because at most one can be chosen from each clique, and there are $k$ cliques, there must have been exactly one chosen in each. By construction, no literal and its negation was chosen. This means we can set the chosen vertices' associated variables to be the truth value for that literal, satisfying each clause.

*No false negatives*: Suppose the 3-Sat instance has a satisfying assignment. Then each clause is satisfied in the 3-Sat instance, given that truth value assignment. We can choose arbitrarily one of the set-to-true literals in each clause and select its vertex in the corresponding clause. No such chosen vertices have an edge to a chosen vertex in any other clique (because the truth value assignment would not set a variable to both true and false). Thus, the graph has a valid independent set, and it is of size $k$ because one was chosen in each of the $k$ independent cliques. Therefore, Independent Set will return true for that graph with the parameter $k$.