### Name: Tradd Schmidt

### This reflection is to be completed individually, though consultations with TAs and classmates are encouraged as long as they are appropriately acknowledged.

### Included in the assignment are some sample interview questions that were culled from the web.

### This reflection document is designed to help you grapple with considering the data structure choices. For each of the five question you decide to try to answer, copy the interview question, explain the data structure and the algorithm you would use to solve the problem (in English, NOT CODE), and finally why you think the data structure you chose is among the most appropriate choices for the problem. In order to answer the last question, it may help to consider alternative data structures that could make trying to solve the problem considerably worse.

1. Copy the question you chose to answer in the box below

|  |
| --- |
| 1. Given a string such as ‘I am a human being’, design an algorithm that would output a string that reverses all the letters in the words, but not the input string itself. The output should be “I ma a namuh gnieb”. |

What is the data structure and algorithm you would use to solve the problem? Use English and NOT CODE.

|  |
| --- |
| I would use lists.  First I would split the string twice so each word has a list containing each letter as an element of the list.  I would create an empty string to append the letters into.  I would then check the length of each word’s list. If it is 1 element long, nothing will be done to it.  I would then iterate through the list backwards and append each letter into the empty string.  When a list finishes I append a space to the empty string. |

Explain why the data structure you designed is among the most appropriate choices, perhaps by comparing it to other alternatives:

|  |
| --- |
| I used a list because for each word I want to start at the end of the list. Lists are indexed and each value is a reference. This makes it easy for the computer to start at the end as opposed to an array for example where I would have to start at the beginning and go through the entire list to reach the final element. |

1. Copy the question you chose to answer in the box below

|  |
| --- |
| 1. On an office floor, there are *e* entities, such as Cubicles, Walls, and Coffee Rooms. Design a data structure to store this information so that when a new member joins the company, an empty cubicle that is closest to any coffee room is assigned to the employee. A person can walk through cubicles to go to the coffee room but not through walls. |

What is the data structure and algorithm you would use to solve the problem? Use English and NOT CODE.

|  |
| --- |
| I would use an array.  The head of the array would be an indicator on whether the cubicle is filled or not. (0 if the cubicle is empty and 1 if it is filled)  Then the subsequent nodes would essentially integers indicating the amount of distance you would have to walk in a direction to move closer to a coffee room. (e.g. node 2 would be 30 ft. and then node 3 would be 20 ft. and then node 4 would be 25 ft.)  The values for each node following the head would be added together so the “length” of the cubicle is known.  Each time a new worker is hired, the array that has the smallest length and is still open is assigned to the new worker and the head of the array’s value is changed to 1. |

Explain why the data structure you designed is among the most appropriate choices, perhaps by comparing it to other alternatives:

|  |
| --- |
| I used an array because the data will all be integers. In addition the distance for each cubicle is a sequential set of instructions that is relevant to the previous distance that was traveled. This is reflected in the nodes of the array for each cubicle. |

1. Copy the question you chose to answer in the box below

|  |
| --- |
| 1. You are given a dictionary with a set of words in it, and you are to group all the words that are anagrams of each other into sets.   For example:  Input: “bat”, “tar”, “art” “xyz” , “design”, “art”, “singed”  Output: ["bat", ["tar", rat", “art”],”xyz”, [“design”, “singed”]] |

What is the data structure and algorithm you would use to solve the problem? Use English and NOT CODE.

|  |
| --- |
| I would use a dictionary. Each key would be a grouping of specific letters and the values for each key would be other words that share the exact same letters.  I would have a loop that iterates through all of the words.  Each word would be checked with the keys of the dictionary to see if it is a unique group of letters.  If there isn’t a key that has the same letters, the current word is added as a new key. |

Explain why the data structure you designed is among the most appropriate choices, perhaps by comparing it to other alternatives:

|  |
| --- |
| A dictionary allows you to group pieces of similar data together under a common unifying factor for each grouping. (each value for each key is related to the key itself). Other data structures don’t give you as much control for grouping things together in this manner. |

1. Copy the question you chose to answer in the box below

|  |
| --- |
| 1. Given a list that can contain duplicate values, output the first unique element in it.   For example:  Input: BH BH Q AL HJ AL HJ PK  Output: Q  Input: BH BH Q AL HJ AL HJ PK Q  Output: PK (OOPS) |

What is the data structure and algorithm you would use to solve the problem? Use English and NOT CODE.

|  |
| --- |
| I would use a list.  I would create an empty list.  Then each time I go to append an item to the list, I will iterate through the list.  If I encounter an element that is identical to the current word I am appending, I will delete that element and break the iteration for that word (thus not adding the current word to the list).  Once all values have been checked there will only be unique elements that were added.  I will then print the first element. |

Explain why the data structure you designed is among the most appropriate choices, perhaps by comparing it to other alternatives:

|  |
| --- |
| I used a list because I will need to iterate through the values, but the values may not be of the same type. In the example they are all strings, but there is a possibility that there could be integers for example. Therefore, we need a list which does not restrict the type of values that are in the data structure. |

1. Copy the question you chose to answer in the box below

|  |
| --- |
| 1. How would you design a cell phone’s contact list that gives you the contacts that have their first or last names that contain certain letters you type in? For example, If you press “M” it will tell you all the contacts starting with “M”. If you then press “MI”, it will tell you all the contacts starting from “MI” and so forth.... |

What is the data structure and algorithm you would use to solve the problem? Use English and NOT CODE.

|  |
| --- |
| I would use an array.  Each contact would be an individual array where each letter is a node and has a link to the next letter in the name of the contact.  When a letter is entered, the head of each array will be checked to see if the value matches.  Each array that has a head node that has the same value of the entered character would be set to a variable and added to a list.  When another character is typed, the list we just created would be iterated through.  Each variable’s (that contains a head node) link would be checked to see if the next value in the array matches the entered character.  If it matches then the variable will be set to the link. If it does not match, then the variable is deleted from the list. |

Explain why the data structure you designed is among the most appropriate choices, perhaps by comparing it to other alternatives:

|  |
| --- |
| An array is most appropriate here because each value that will be checked is dependent on the value before it. All of the values will also only be strings because they are names. |