Elementary Data Structures and Algorithms

Searching

Concept: searching

1. **T** or **F**: The following code reliably sets the variable *min* to the minimum value of an unsorted, non-empty array.

```
min = 0;
for (i from 0 until array.length)
    if (array[i] < min)
        min = array[i];
```

2. **T** or **F**: The following code reliably sets the variable *max* to the maximum value in an unsorted, non-empty array.

```
max = array[0]
for (i from 0 to array.length)
    if (array[i] > max)
        max = array[i]
```

3. **T** or **F**: The following function reliably returns `True` if the value of item is *present* in the unsorted, non-empty array.

```
function find(array,item)
    {
    found = False;
    for (i from 0 until array.length)
        if (array[i] == item)
            found = True;
    return found;
    }
```

4. **T** or **F**: The following function reliably returns `False` if the value of item is *missing* in the unsorted, non-empty array.

```
function find(array,item)
    {
    found = True;
    for (i from 0 unitl array.length)
        if (array[i] != item)
            found = False;
    return found;
    }
```

5. What is the average and worst case time complexity, respectively, for searching an unordered list?

    A. linear, linear

    B. linear, log,

    C. log, linear

    D. log, log

6. What is the average and worst case time complexity, respectively, for searching an ordered list?

A. log, log

B. linear, log

C. linear, linear

D. log, linear