1) Modify the following to correctly grow the region of the triangles t marked (t.m==true) assuming that initially one vertex v is marked (v.m==true) and using markings on both vertices and triangles to grow concentric layers (rings) of triangles around V. The three vertices of triangle t are (t.a, t.b, t.c).

```
Repeat L times {
    For each triangle t {
        if (t.a.m || t.b.m || t.c.m) {
            t.m=true;
            t.a.m=true; t.b.m=true; t.c.m=true; # remove this line
        }
    }
    For each triangle t {if (t.m) {t.a.m=true; t.b.m=true; t.c.m=true;}} # add this
```

2) Assume that you are using a **binary search** to find the index of the entry in table T that has value V

What assumption about T must hold? **T must be sorted**

If T has 1000 elements, how many tests will you need? **10 or 11**

3) The pseudo-code below has a bug.

```
for (i=0; i<n-1; i++) {
    for (j=0; j<n-1-i; j++) {
        if (T[j+1] < T[j]) { # insert “temp=T[j];”
            T[j]=T[j+1];
            T[j+1]=T[j]; # replace by “T[j+1]=temp;”
        }
    }
}
```

What was it supposed to do? **Bubble Sort**

What is the bug? **Swap is overwriting itself**

Indicate how to fix it by marking the code above.

4) You know that algorithm X(n) is O(log(n)). X(u) took 1mn. X(v) took 2 mns.

I conclude that v=2u. Am I right? **No**

Justify my conclusion if correct or rectify it if wrong? **v=u²**

5) Explain what a greedy algorithm is.
An algorithm that attempts to minimize some global cost $C$ by selecting at each step the move that minimizes $C$ (the locally optimal choice), without looking ahead to select the globally optimal sequence of moves.