Linear Time vs Quadratic time

1) A language is decidable in linear time if its time complexity is O(n). To show this we will construct an algorithm that will return the time complexity for all languages $L_1, L_2, L_3 \ldots L_m$ in time mn. We will use C++ for this. Languages $L_1, L_2, L_3 \ldots L_m$ are stored in vector arr. The input string is held in a string called input.

```
int func(string input, vector<language> arr){
    int i=1;
    int n= input.length();
    while(i< arr.size ()){
        i++;
    }
    return (n* i);
}
```

Since this algorithm visits each element once it is done in O(n) time. This means that the language is decided in linear time.

Linear vs Quadratic time 2

LT is a proper subset of QT because all languages that can be decided in $O(n)$ can be decided in $O(n^2)$. This is accomplished by putting the previous function in a for loop

```
for(int j = 1; j < arr.length(); j++){
    // func loop
}
```

This will make the function go through the array m times meaning it checks m^2 elements. This means its time complexity is $O(n^2)$.

However although all linear time languages can be solved in quadratic time, There are algorithms that can only be solved in quadratic time. One example is bubble sort.

→

flip page

```
void swap(int &a, int &b){
    int tmp;
    tmp = a;
    a = b;
    b = tmp;
}

void bubbleSort(vector<int> arr){
    for(int i=0; i < arr.size(); i++){
        for(int j=0; i < arr.size(); j++){
            if(arr[j] > arr[j+1]){
                swap(arr[j], arr[j+1]);
            }
        }
    }
```

Since all Languages in LT are in QT but not all Languages in QT are in LT. LT is a proper subset of QT.

Polynomial Time complexity

We proved in the previous problem that a problem solved in $O(n)$ can be solved in $O(n^2)$ by adding an extra for loop that does nothing but make it loop n more times. This means that for any problem solved in $O(n^k)$ can be solved in $O(n^{k+1})$ by adding a useless loop. Therefore as long as one problem solved in $O(n^{k+1})$ cannot be solved in $O(n^k)$, $O(n^k)$ is a proper subset of $O(n^{k+1})$.