

When analyzing the asymptotic time complexity of an algorithm, we ignore the actual amount of time it takes for a given statement of code to execute. Explain why we are able to do this and still get useful information about the execution time of an algorithm.

Give the asymptotic time complexity for the following function.

```
int quiz3(int i, int j) {  
2   while (i != j) {  
    if(i<j) {  
4       j--;  
    } else {  
6       return quiz3(i, j+1);  
    }  
8   }  
    return 0;  
10 }
```



The Standard Template Library provides a number of generic algorithms. What advantages do they provide and what mechanisms do they use to achieve these advantages?

Give an example of an application in which an STL `set` would be a good choice for storing data. Give an example of an application for which a container other than an STL `set` would be a better choice. Be sure to justify your reasoning.

Given the `LstIterator` and `Item` classes definition below, implement the increment operator for the `LstIterator` class.

```
template<class T>
class Item {
    friend class List<T>;
    friend class LstIterator<T>;
protected:
    Item(const T& val=T());
private:
    T ItemValue;
    Item* Next;
};

template<class T>
class LstIterator {
    friend class List<T>;
public:
    T& operator*();
    LstIterator& operator++(); // <-- implement this one
private:
    Item<T>* FirstItem; // pointer to current element
};
```



Briefly explain the differences between the following two member functions from the STL multiset class: `lower_bound()` and `upper_bound()`.