Linear Linked List
Templated, will hold 2-3 trees

```
public:
        insert(data_type & insert)
        remove(data_type & find_and_remove)
        get(data_type & find)
        ...
        virtual operator[](const int); // calls get
        virtual operator+=(const data_type);
private:
        lll_node<data_type> head
```

2 - 3 tree
Templated, will hold tasks

```
public:
        insert(data_type & insert)
        remove(data_type & find_and_remove)
        get(data_type & find)
        …
        virtual operator[](const int); // calls get
        virtual operator+=(const data_type);
private:
        two_three_node<data_type> head
```

Study_Guide
Derives from lll<two_three<task>>

```
public:
        // Adds the task to the correct tree
based on its category
        operator+=(const task);
        // Display all the tasks by category
        friend ostream & <<(c ostr &, c s_g &);
```

Task
Hold information on what to study + priority

```
public:
        // Display the task
        friend ostream & <<(c ostr &, c task &);
        // Take in a task
        friend istream & >>(c istr &, c task &);
        // The long list of comparison operators
private:
        // All the data we need
        Int priority;
        Char * category;
        Char * description;
        Bool completed;
```