

PRACTICE MIDTERM EXAM #1 - SOLUTIONS

1. ADTs

```
static void printCharPairFrequencies(const Lexicon& english) {  
    Map<string,int> pairs;  
    for (string word : english) {  
        for (int i = 0; i < word.length() - 1; i++) {  
            pairs[word.substr(i,2)]++;  
        }  
    }  
    for (string pair : pairs) {  
        cout << pair << " " << pairs[pair] << endl;  
    }  
}
```

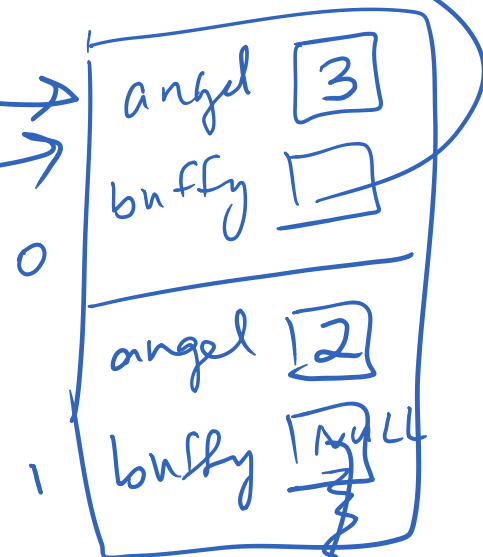
2. Pointers and Memory

DRAWING:

Stack:



Heap:



3. Recursion [by Marty Stepp]

```
string moveToEnd(string s, char c) {
    if (s.length() == 0) {
        return "";
    } else if (toupper(s[0]) == toupper(c)) {
        char upperC = toupper(s[0]);
        return moveToEnd(s.substr(1), c) + upperC;
    } else {
        return s[0] + moveToEnd(s.substr(1), c);
    }
}
```

```
string moveToEnd(string s, char c) {
    if (s == "") {
        return s;
    }
    char first = s.at(0);
    s = s.substr(1, s.length() - 1);
    s = moveToEnd(s, c);
    if (toupper(first) == toupper(c)) {
        char upperFirst = toupper(first);
        return s + upperFirst;
    } else {
        return first + s;
    }
}
```

4. Classes

a)

private:

```
    int startTime;
    int durationInMinutes;
    DayPattern daysOfWeek;
};
```

b)

```
Lecture::Lecture(DayPattern daysOfWeek, int startTime, int duration) {
    this->daysOfWeek = daysOfWeek;
    this->startTime = startTime;
    this->duration = duration;
}
```

```
int Lecture::startTime() {
```

```
    return this->startTime;
}

int Lecture::durationInMinutes() const {
    return this->duration;
}

DayPattern Lecture::daysOfWeek() const {
    return this->daysOfWeek;
}
```

c)

```
int Lecture::endTime() const {
    int startHour = startTime / 100;
    int startMinute = startTime % 100;
    int endHour = (duration + startMinute) / 60 + startHour;
    int endMinute = (startMinute + duration) % 60;
    return 100 * endHour + endMinute;
}
```

5. Big-O

O(1)

O(NlogN)

O(N²)