CS 106B Section 4 (Week 5) Solutions

---

1. partitionable

```
bool partitionable(Vector<int>& list) {
    return helper(list, 0, 0);
}

bool helper(Vector<int>& rest, int sum1, int sum2) {
    if (rest.isEmpty()) {
        return sum1 == sum2;
    } else {
        int n = rest[0];
        rest.remove(0);

        bool answer = helper(rest, sum1 + n, sum2) ||
                      helper(rest, sum1, sum2 + n);

        rest.insert(0, n);
        return answer;
    }
}
```

---

2. makeChange

```
void makeChange(int amount, Vector<int>& coins, Vector<int>& chosen) {
    if (coins.isEmpty()) {
        if (amount == 0) {
            cout << chosen << endl;
        }
    } else {
        int coin = coins[0];
        coins.remove(0);
        for (int i = 0; i <= (amount / coin); i++) {
            chosen += i;
            makeChange(amount - (i * coin), coins, chosen);
            chosen.remove(chosen.size() - 1);
        }
        coins.insert(0, coin);
    }
}

void makeChange(int amount, Vector<int>& coins) {
    Vector<int> chosen;
    makeChange(amount, coins, chosen);
}
```

---

3. longestCommonSubsequence

```
string longestCommonSubsequence(string s1, string s2) {
    if (s1.length() == 0 || s2.length() == 0) {
        return "";
    } else if (s1[0] == s2[0]) {
        return s1[0] + longestCommonSubsequence(s1.substr(1), s2.substr(1));
    } else {
        string choice1 = longestCommonSubsequence(s1, s2.substr(1));
        string choice2 = longestCommonSubsequence(s1.substr(1), s2);
        if (choice1.length() >= choice2.length()) {
            return choice1;
        } else {
            return choice2;
        }
    }
}
```

CS 106B Section 4 (Week 5) Solutions

---

4. Debugging

We never “unchoose”. The line **`numbers.insert(0, first);`** needs to be inserted right before the return.

---

5. Big O

a) O(1)

b) O(N)

c) O(N^2)