March JDCC Solutions + Comments

Problem A (Word Sums):

Solution: Simply loop over the words and add each letter with using modular arithmetic. Note that you can do arithmetic on characters, i.e. if you are adding two characters a and b, you can do:

```
char sum = ( (a - ‘A’) + (b - ‘A’ + 1) ) % 26 + ‘A’;
```

Time Complexity: O(N)

Space Complexity: O(N)

Problem B (Scientific Notation):

Solution: This problem tests basic knowledge of string parsing and rounding. It is important to know these two concepts, and in such problems identifying key edge cases to test (e.g. converting numbers less than 10, or converting 10^9).

Time Complexity: O(log N)

Space Complexity: O(1)

Problem C (Careless Sums):

Solution: This is an ad-hoc number theory problem that is based on the XOR operator. As with the XOR operator, the addition in this problem is an involution, which means that N + K + K = N for all N and K. With this in mind, the best way to solve this problem would realize that summing the numbers between N and M is equal to summing the numbers from 1 to M and adding the the sum from 1 to N-1. With some testing, one can find that if K is divisible by 20, then the sum from 1 to K is equal to K. Using this property, at most 20 additions are required.

Time Complexity: O(*log N*)

Space Complexity: O(*log N*)

Problem D (Space Invaders):

Solution: This problem is solved by binary search on the time needed. For each time, compute how the interval the invaders will conquer, then sort the intervals and check if their union covers the planet. If yes, then binary search down, otherwise up.

Time Complexity: O(Nlog N log T)*,* where T is an upper bound on the time.

Space Complexity: O(*N)*

Problem E (Victory Towers):

Solution: This problem is a spin on the classic problem of “given a list of numbers, split them into two equal groups.” That problem is solving with DP on the number of elements used and the size of one of the groups, while this problem uses DP on hte number of elements used and the size of two of the towers. The transition state is to add the current element to one of the two towers being stored, or discard it (equivalent to adding it to the third tower).

Time Complexity: O(NM^2*)*

Space Complexity: O(NM^2*)*