Code Jam 2011 - Round 3

Analysis: Dire Straights

Dire Straights had one of the easiest small datasets for round 3. Several brute force or backtracking solutions that followed the rules could come up with a valid answer in the allotted time, so rather than look at the small dataset, we will instead examine the large. The large dataset requires some insight into a greedy approach.

For a problem like this, a good strategy is to think of how you might try to solve this problem by hand. A very intuitive strategy is to first put all the cards in order, then start setting the cards down on the table, creating a new straight whenever necessary. Since our goal is to make the length of the shortest straight as long as possible, then one idea that seems like it might work is to always increase the length of the shortest straight when we have a choice. Now we need only to prove that this choice is optimal.

Suppose we have two straights, one from **a** to **b**, another from **c** to **d**, such that **a** < **c** <= **d** < **b**.

```
a-------b
 c-----d
```

Notice that we can replace these two straights with straights from **a** to **d** and **c** to **b** (this change is illustrated below), and this does not decrease the score. In fact, this change has the potential to increase the score.

```
a------d
 c------b
```

This shows that we can always make sure that a straight that started later never ends before one that started earlier. Hence, attaching the next card to the shortest straight is optimal.

Finally, the size of every straight is examined and the length of the shortest straight is the total score achieved in Dire Straights.