Programming Assignment: Poker Sort

Due Date: November 7, 2023

You are to write a function to sort an array of five-card poker hands encoded as integers:

```
void poker_sort(vector<int>& a);
```

Since this is a sorting assignment, you must write your own sorting function. **YOU MAY NOT USE A LIBRARY SORT!!!** Also, you must write your own comparison function (possibly the more challenging part of the assignment! See below). Your score will be based mostly on correctness, but execution speed will also play a role. Specifically, 20% of the score for each case will be the ratio of the fastest time to your time (if your answer is invalid, you get a zero for that case).

Each card will be encoded as an integer 0 ≤ c ≤ 51 as follows: The possible ranks of a card {2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King, Ace} will be mapped in order to {0, 1, …, 12}. The possible suits {Clubs, Diamonds, Hearts, Spades} will be mapped to {0, 1, 2, 3}. Then, if the mapped rank and suit of a card are r and s, respectively, the encoded value will be c = 4 * r + s. Now, given the five cards {c0, c1, c2, c3, c4}, the hand will be encoded as (c0 + c1 * 52 + c2 * 52^2 + c3 * 52^3 + c4 * 52^4) (this fits comfortably in a 4-byte int). Note that though the order of the cards is irrelevant in poker, the encoding *does* depend on the order of the cards.

Given two poker hands h0 and h1, we say that h0 < h1 if h1 beats h0 in a poker game. Notice that there are many hands that tie (although not very common in actual games, it does happen!) Now we need to know the bare minimum of poker rules to be able to determine which of two hands wins.

There are nine types of poker hands. Given any two hands, the higher type always wins. What if two hands are of the same type? Well, it depends. First of all, for this assignment, note that each hand is generated by (virtually) shuffling a standard deck of 52 cards and dealing 5 cards. Thus it is perfectly possible to have two hands both having four 2's (even though there are a total of four 2's in a deck). Notice also that the only role Suit plays is to determine if a hand is a flush; it plays no role in determining which hand is higher (unlike Bridge or Spades or Hearts, etc). Below, the possible types of five-card hands (in descending order of value) are given, and within each type the rules for determining the order of hands of that type. In all cases, if the given criteria fail to determine a winner, the hands tie.

Straight Flush - a straight and a flush simultaneously (see below for those terms). The hand with the highest card wins.

Four of a Kind - 4 cards have the same rank. If the ranks of the 4 equal cards in the two hands, differ, the higher wins, otherwise the higher fifth card wins.

Full House - three of a kind and a pair. The higher 3 equal-ranked cards wins, then the higher pair.

Flush - all 5 cards have the same suit. Sort the cards from highest to lowest, and then the lexicographically higher hand wins.

Straight - the 5 cards are consecutive. Note that here there are some idiosyncracies, namely, an Ace can be either the low card or the high card, i.e., {Ace, 2, 3, 4, 5} and {10, Jack, Queen, King, Ace} are both straights, but {King, Ace, 2, 3, 4} is NOT a straight. The highest card of the straight determines the winner.

Three of a Kind - 3 cards have the same rank. The higher 3 equal-ranked cards wins, otherwise the remaining two cards are compared lexicographically.

Two Pairs - there are two distinct pairs. The higher pair wins, then the lower pair, then the fifth card.

Pair - 2 cards have the same rank. The higher equally-ranked card wins, otherwise the remaining 3 cards are compared lexicographically.

High Card - none of the above. The hands are compared lexicographically.