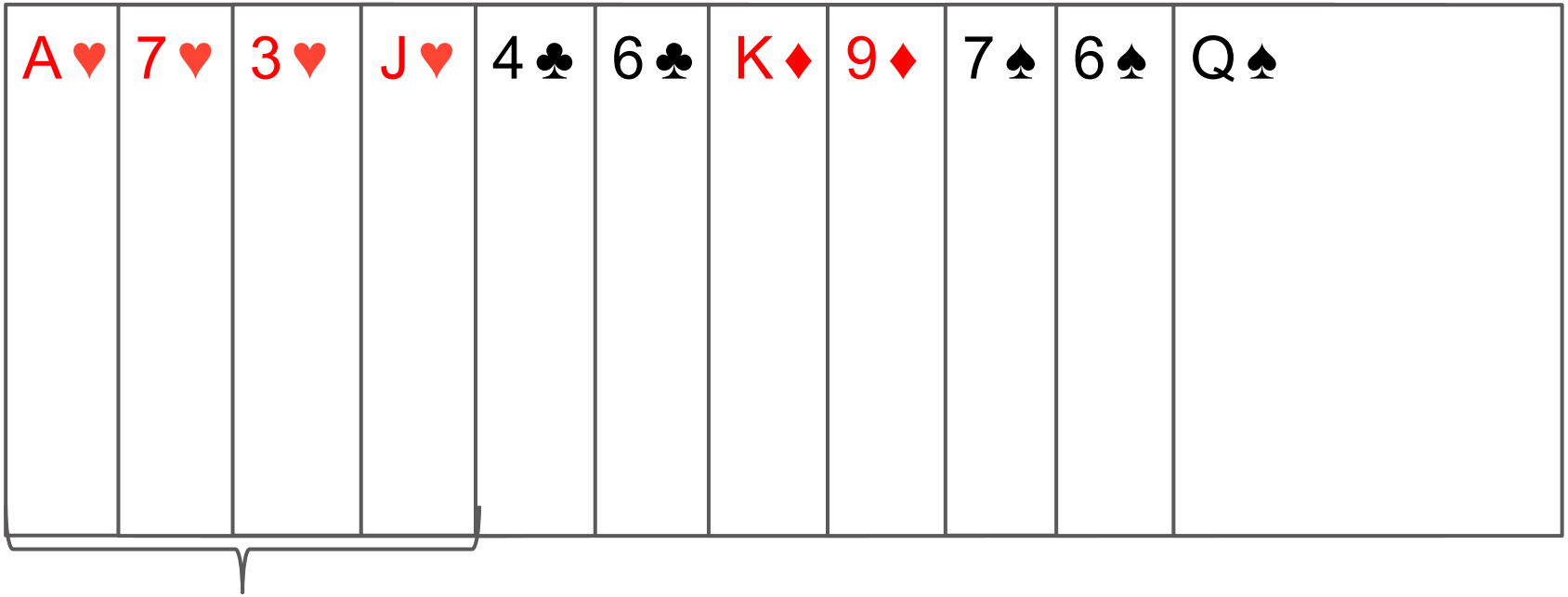
Hearts

Kevin Miller
(kmiller4@stanford.edu)

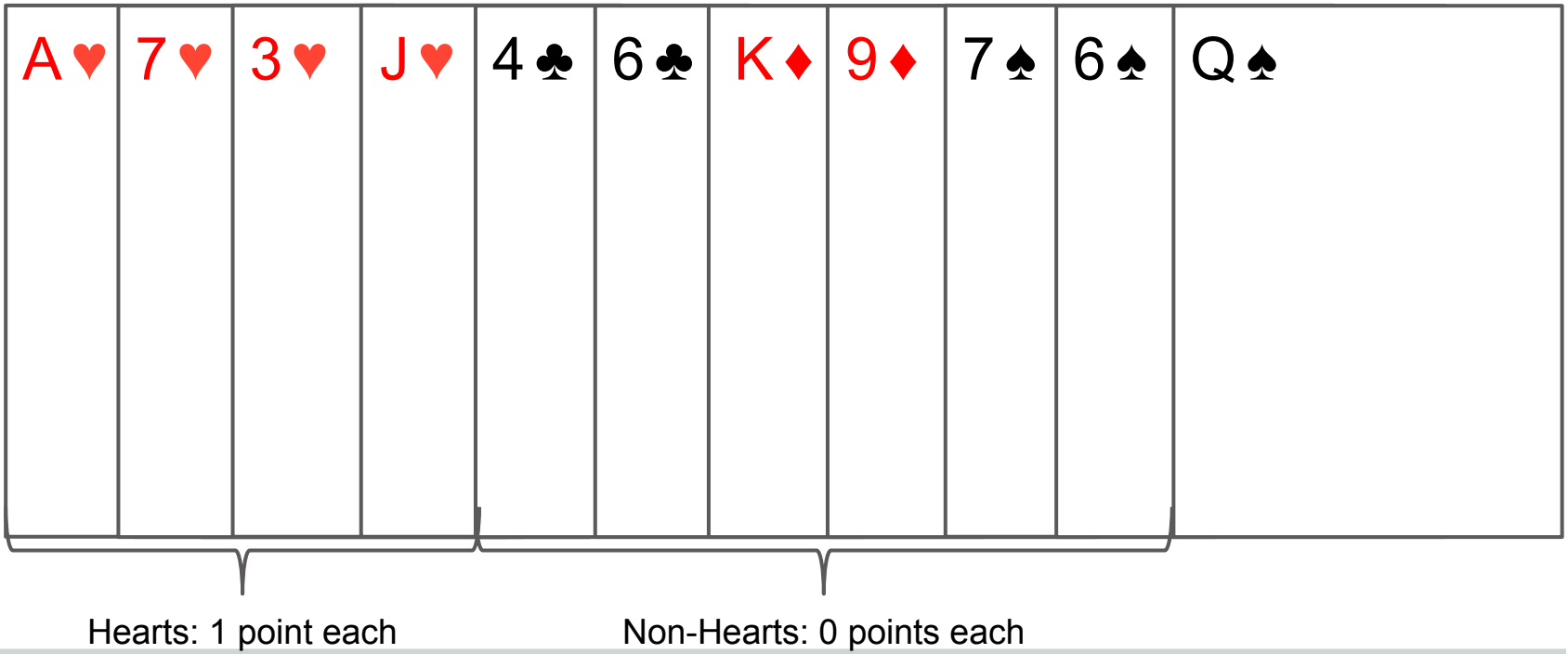
Administrivia

- GraphViz due tonight at midnight or tomorrow at midnight with 1 late day
- Office Hours: Tonight 6-9
- Email cs106l-aut1314-staff@lists.stanford.edu for questions

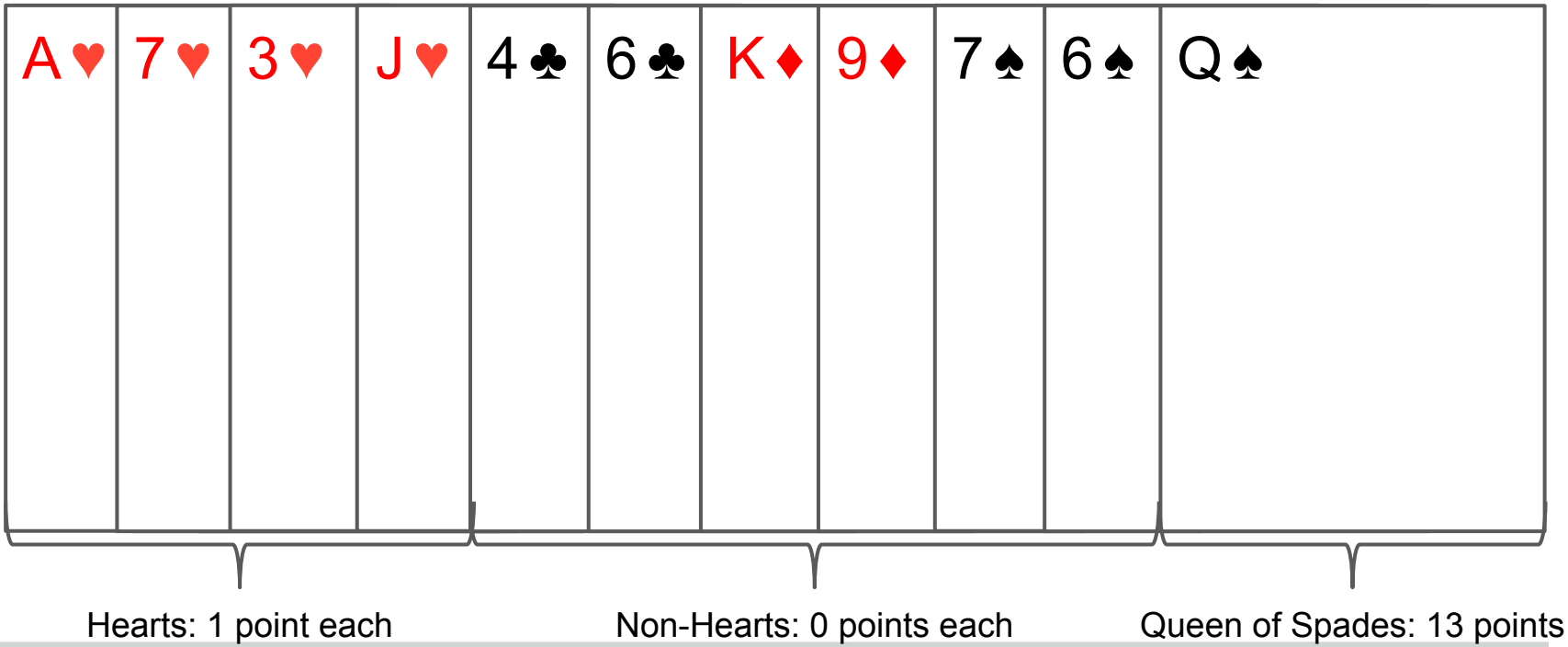
How to Play Hearts



How to Play Hearts



How to Play Hearts



How to Play Hearts

1. Leader plays card
2. Everyone else matches suit if possible
3. Highest value of leader’s suit wins
4. Winner becomes new leader

Key Point

Ordering of players is preserved

- Who goes first changes but play always progresses counter clockwise

Terminology

- Trick: a single event where each player plays a card and the winner takes them
- Round: 13 tricks
- Leader: person who plays first card of the trick
- Match Suit: playing a card of the same suit as the leader

Strategy

- Leader plays the lowest card they have left
- Others plays lowest card possible if they can match suit or highest card otherwise

Quick Enumeration

An enumerated type is an int that takes on a specific range and each value has a name

Allows a variable to be a set of predefined constants

Quick Enumeration

```
enum Direction {
   LEFT, //has value 0
   RIGHT //has value 1
};

Direction zoolander = RIGHT;
int n = 2 * zoolander; //n==2
```

Let’s Do It!

Coding Time!

random_shuffle

Parameters: iterator begin, iterator end

Return: none

Function: shuffles the elements between begin and end

find_if

Parameters: iterator begin, iterator end, bool function isValid

Return: iterator to the first element matching comparator or end if no element is found

Function: finds an element that isValid if it exists

copy

Parameters: iterator begin, iterator end, iterator copy location

Return: none

Function: copies the elements from begin to end to location

copy

- Random access iterators can be incremented using + and +=
- same as doing ++ a set number of times in a for loop
- makes accessing particular elements easy

sort

Parameters: iterator begin, iterator end, bool comparator function

Return: none

Function: sorts the elements from begin in ascending order

Comparator Function

- Many algorithms and containers have a concept of order
- Ascending order → n1 < n2 < …
- A Comparator Function in C++ is a function that determines whether one element is less than another
- Equivalent to implementing the ‘<’ operator

min_element/max_element

Parameters: iterator begin, iterator end, bool comparator function

Return: iterator to the min/max element

Function: finds the min/max element between begin and end

equal_range

Parameters: iterator begin, iterator end, v, bool comparator function

Return: a pair of iterators denoting the begin and end of a range of equal value or end if no range exists

Function: similar to lower_bound from city finder. Finds a range that has a certain value

rotate

Parameters: iterator begin, iterator middle, iterator end

Return: none

Function: rotates the collection such that middle is now first while preserving ordering