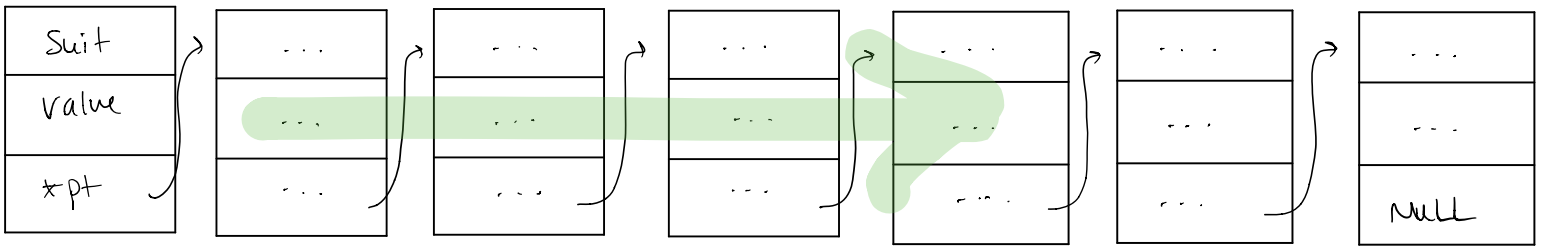
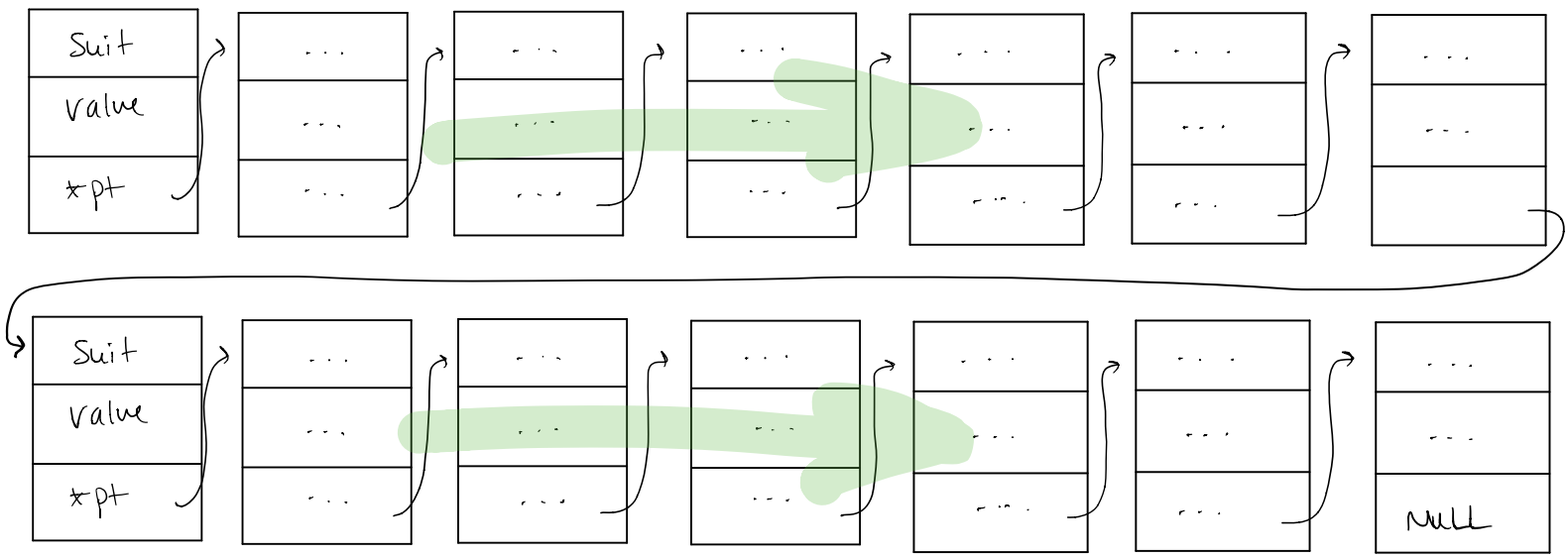
Singly-Linked List Implementation

Player 1's Hand (7 cards initially)

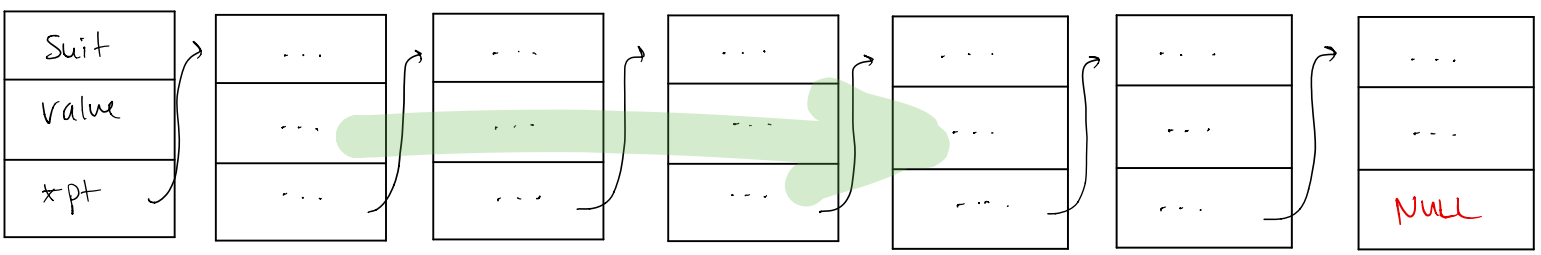


Deck of cards (Pool of remaining cards)



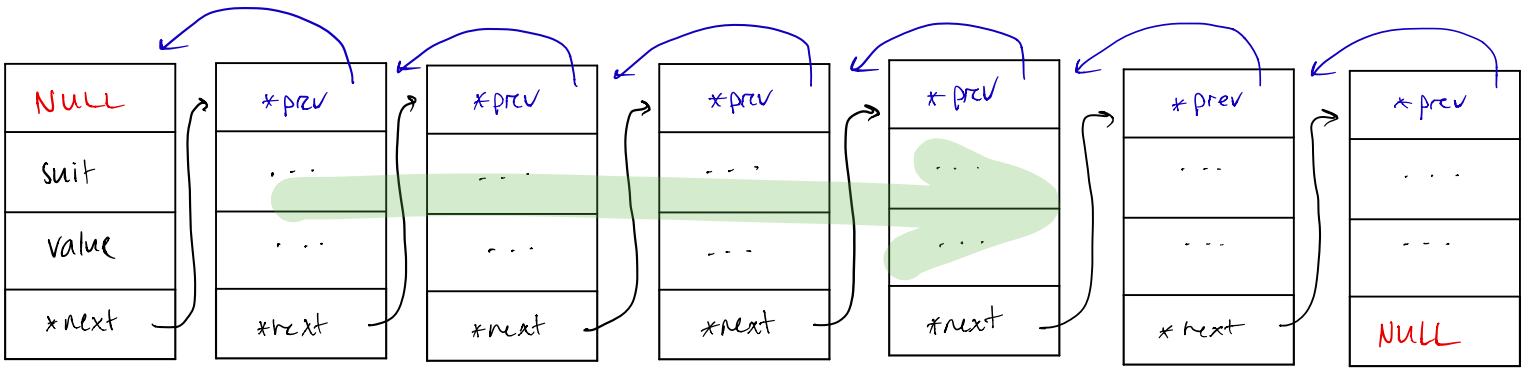
: etc

Player 2's Hand (7 card initially)

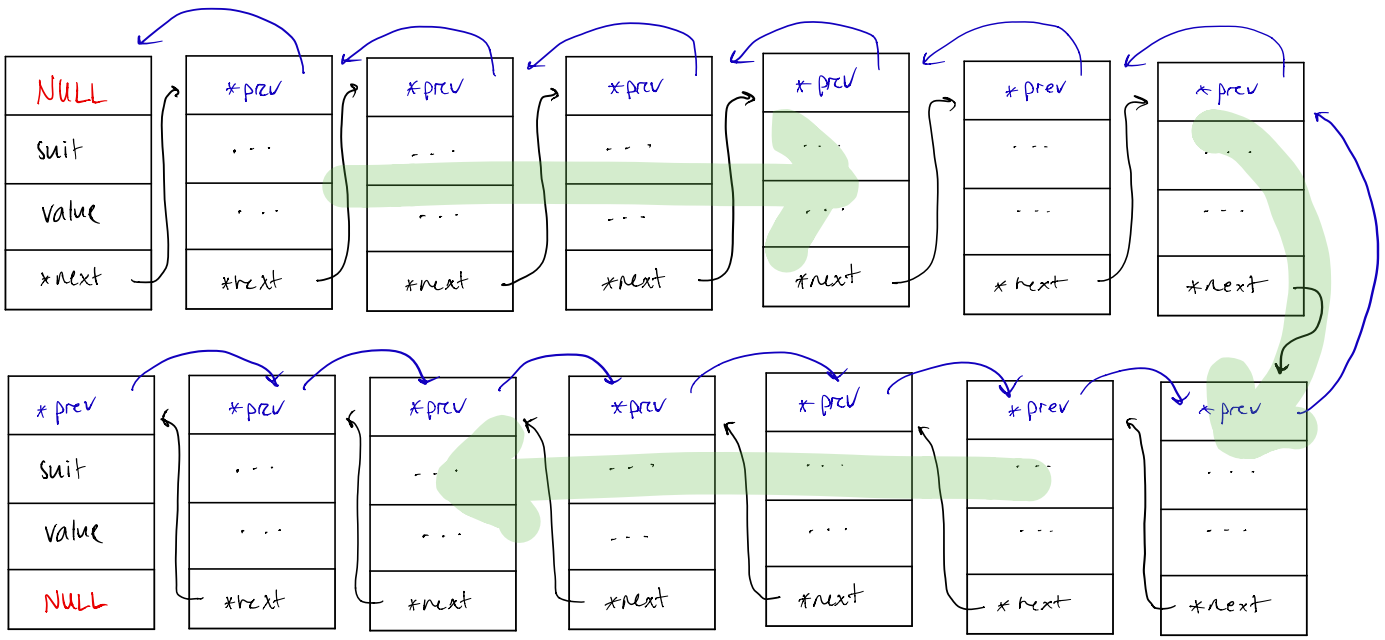


Doubly-Linked List Implementation

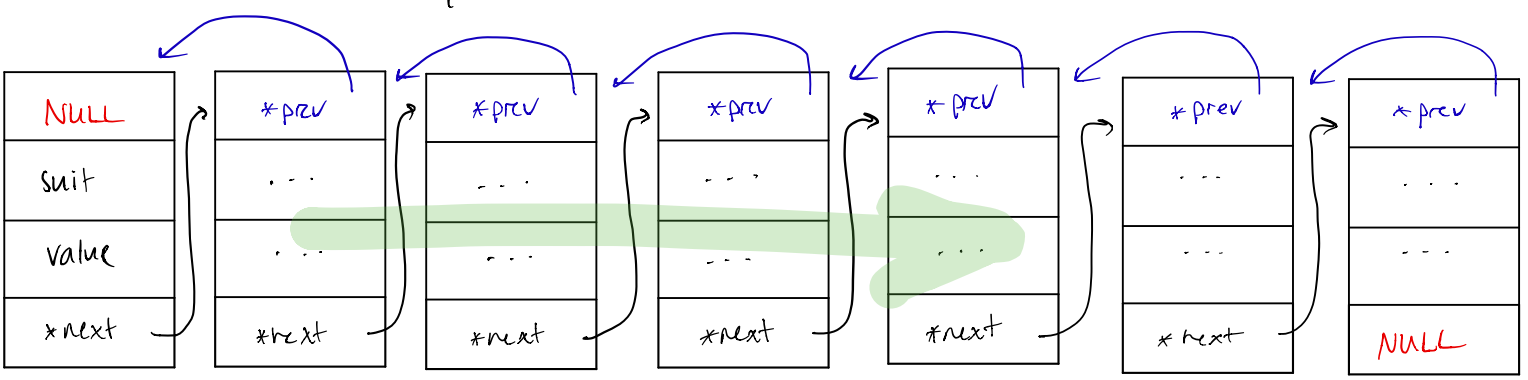
Player 1's Hand (7 cards initially)

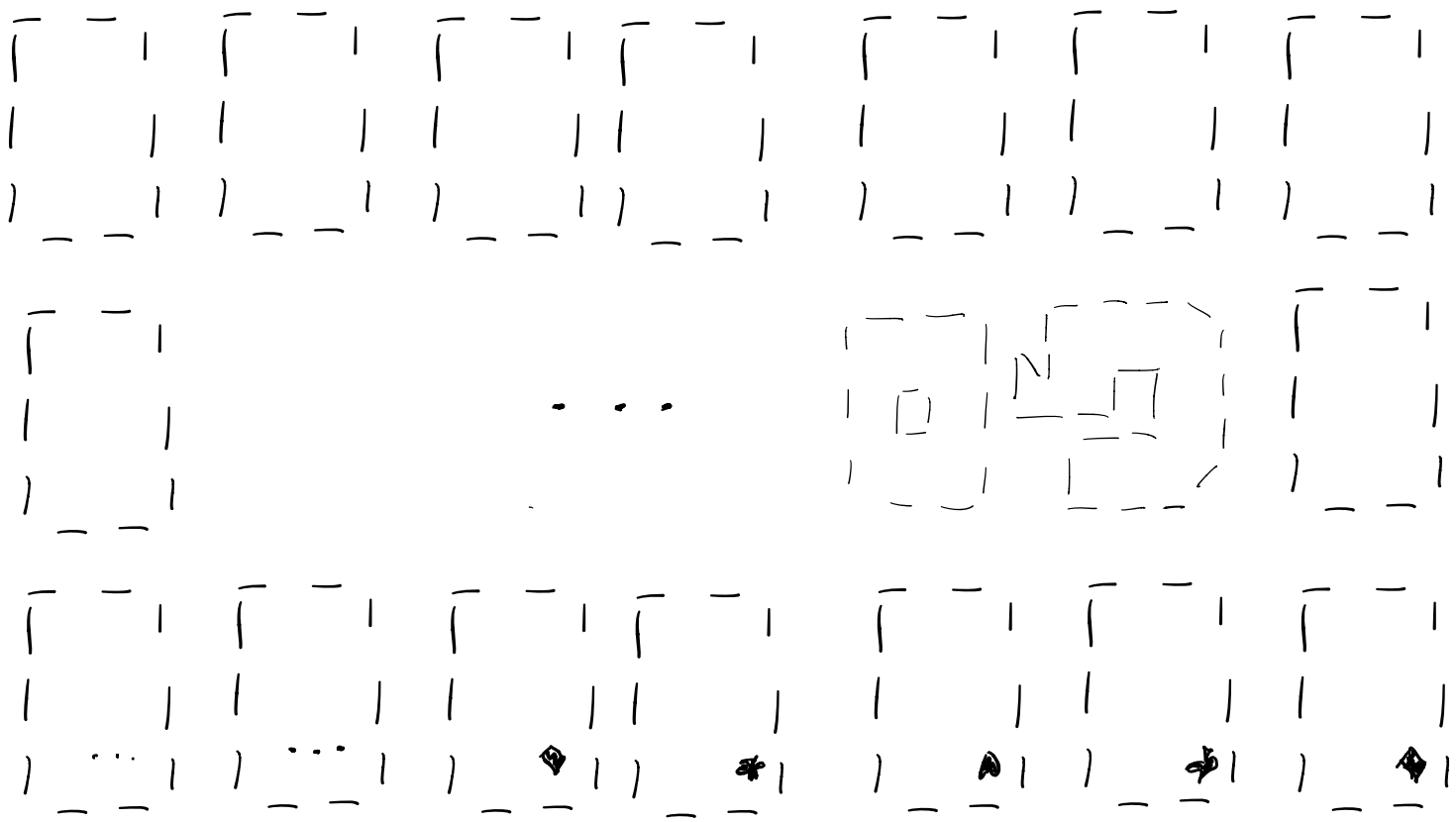


Deck of Cards (Pool of Remaining Cards)



Player 2's Hand (7 cards initially)





Rules : Two player

- 7 cards each initially, pool of shuffled cards in between.
- Player 1 asks player 2 for rank of card, if any match, Player 1 receives the cards (all) else, "Go Fish" & pulls card from pool.
- If a player runs out of cards, they are the winner.

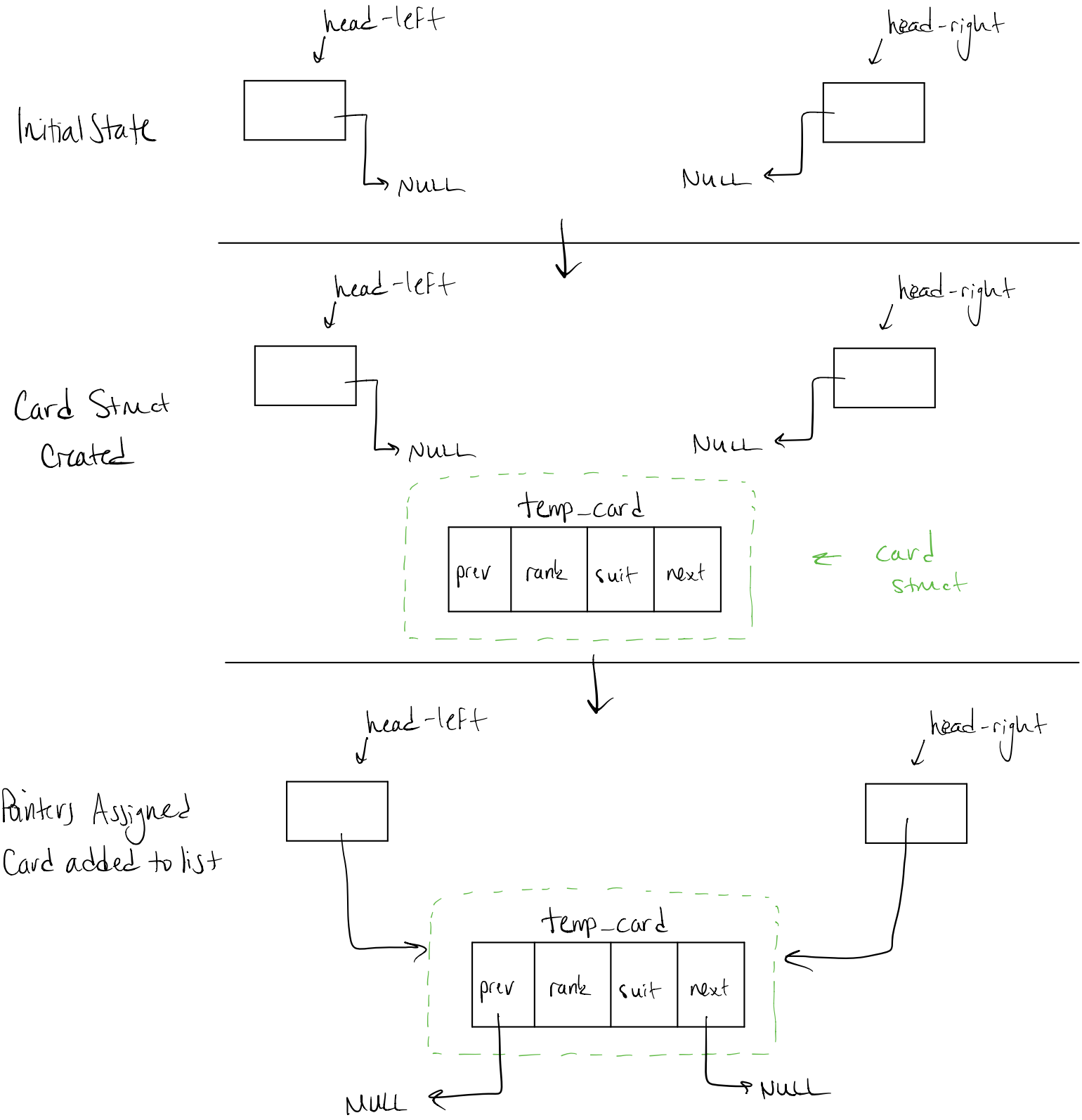
General Structure of the Program

1. Initialize deck from file
2. Shuffle deck
3. Disperse cards to players hands (7 each, 1 at a time)
4. Begin game starting w/ player 1
5. Player 1 requests for a card rank that currently exists in their hand
6. check if Player 2 has that rank (all suits)
   - if so, give them to Player 1
   - else, go fish & pull cards from pool of cards.
7. transfer turn to player 2, repeat step 6 as player 2
8. Check if the cards pulled from player into hand completes a book of cards.
   - if so, remove book from players hand as complete
   - else, continue game
9. Continue process until one of the players has no more cards

possible Functions:

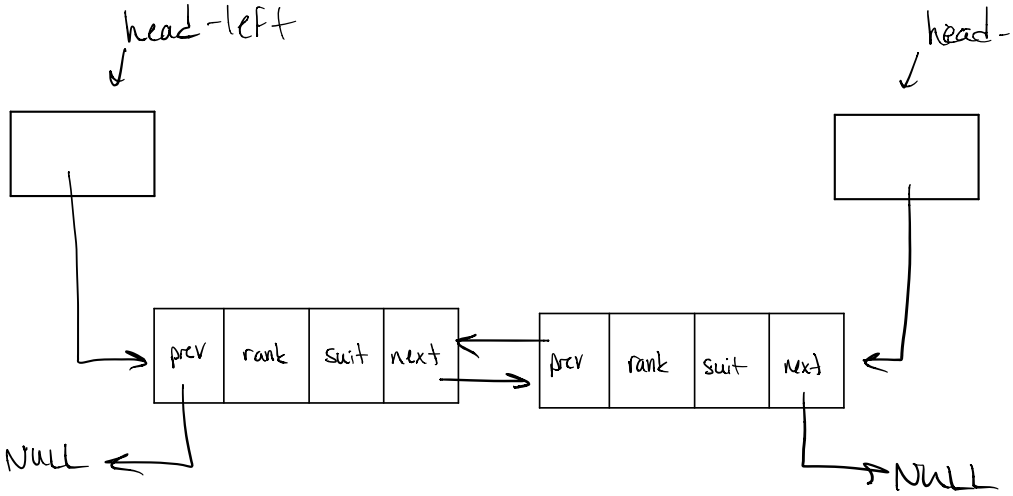
```
Void  create_list ()      // reads File & populates Linked List
Void  add_to_end ()       // used by above
Cards* pull_card_data ()  // parses line from File & returns card struct.
int   find_length ()      // finds length of L.L.
void  print_list ()       // print output
Void  print_formatted_list () // print formatted output
void  shuffle_cards ()    // shuffle deck (x number of swaps)
void  swap ()             // used to swap values of cards, essentially
                             shuffling.

void  generate_playerdeck (card *hl, card **phl);
      // This function will pull x amount of cards from
      // the central deck & populate player deck
```

Adding to Empty Doubly-Linked List

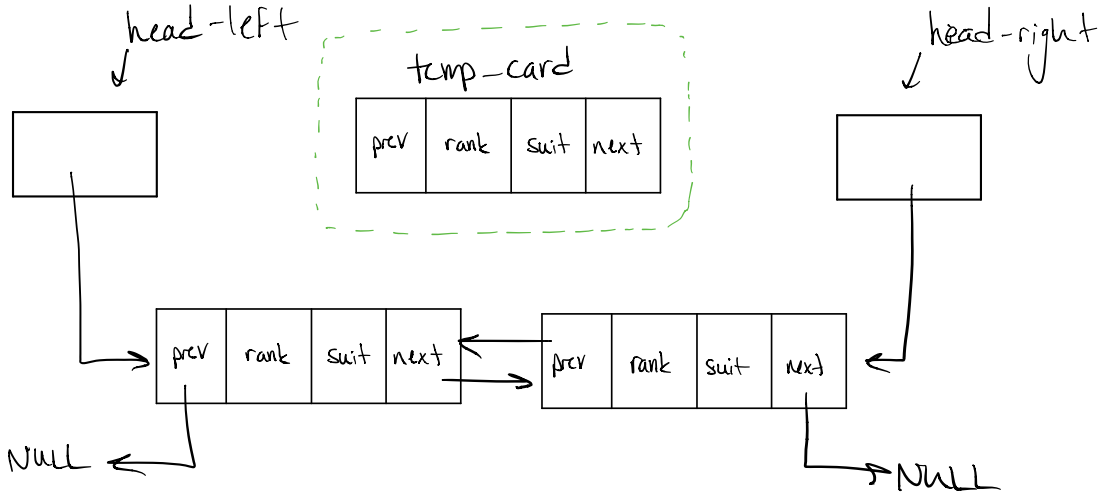


Adding to Doubly-Linked List (Non-empty)

Initial State



Card Struct Created.



Card Added to the end.