



A new course announcement has been posted. [here](#) to view the new announcement.  
Click

[Course](#) > [Modul...](#) > [Final Pr...](#) > Bridge ...

## Bridge - Bidding

WARNING: For the purposes of Academic Integrity, the final project is treated like a final exam. This means penalties are increased for any academic offences (including, but not limited to receiving 0 in the entire course!) You have been warned.

**Reminder:** Do not discuss this assignment with anyone except the instructors on the EdX discussion forums!

### Bidding

As mentioned in the previous problem, Bridge is a game played with 4 players. The players are traditionally denoted using the cardinal directions. There is a North player, an East player, a South player and a West player. The North and South players are partners as are the East and West players.

Order of play begins with a starting player and cyclically goes through the players in the order North, East, South, West, North, East, ... and so on. All 52 cards are distributed evenly amongst the players and the dealer begins play.

The first phase of bridge involves bidding. For the purposes of this project, we will not implement any conventions but rather make a bidding system that accurately reflects what can be done in the game. There are a few different kinds of bids that can be made:

- **pass.**
- a number between 1 and 7 followed by one of the four suits or "no trump" (such as 1♣ or 5NT) called a **numeric bid**.

- a **double** of a previous opponents bid.
- a **redouble** of a previous opponents doubled bid.

A **pass** bid means that you do not think your team could make a contract higher than the most recent bid. If all four players pass during the initial contract bidding phase, the cards are redealt and a new hand is played. Otherwise, bidding continues until there are three consecutive passes. The last non-passed numeric bid (along with any doubling or redoubling) becomes the **contract**.

## Numeric Bids

Numeric bids are bids consisting of a number and a suit, such as **1♣**. Simplifying Bridge slightly here for those that know the game, the bid **1♣** suggest to your opponents that you think your team can make  $6 + 1$  tricks with **clubs** as the **trump suit** (more on trump suits in the playing section of the game). Notice that since there are 13 tricks, bidding anything means you believe your team can take the majority of the 13 total tricks. Subsequent numeric bids must be higher than previous numeric bids. The order of bids is determined first by number and then by suit. So a bid with a higher number is higher. Otherwise, two bids with the same number are distinguished by the following (alphabetic) order of the four suits: ♣, ♦, ♥, ♠ and lastly, "no trump" denoted by NT. As examples, **1♣** is less than **1♦**, which is less than **1NT**, which is less than **2♥** et cetera. We define a **contract** as the last non-passed numeric bid (quantified by doubles or redoubles as necessary).

## Double Bids

Once a numeric bid is made by a team, the opposing team may **double** this bid. A double can only be made if the most recent non-passed bid belongs to the opposing team. For example, if North bids **1♣**, then East may **double** this bid or, assuming East and South pass, West may also **double** this bid. South cannot double his partner's bid. When a contract ends, if the last non-passed bid was a **double** then the points roughly speaking will be doubled (more in this in scoring).

## Redouble Bids

Similarly, a **redouble** can only be made if the last bid by the opposing team was a **double** and there are no other bids in between. So for example, if North bids **1♣** and East **doubles** the bid, then South can **redouble** or, assuming both

South and West pass, North can also **redouble** the bid. If a contract ends in a redouble, then the points roughly speaking will be quadrupled (more on this in scoring).

## Winning the Contract

The team who won the contract is the team who made the last numeric bid. Suppose bidding ends and "North" is the last player to make a numeric bid. This bid represents the contract. Any **doubles** or **redoubles** on this final numeric bid is also included in the final contract (more on this in scoring).

If the North and South team win the contract, then the first person of North or South to bid the suit named in the winning contract is deemed the declarer and their partner is deemed the dummy (more on this in the playing section). Similarly for East and West.

Here is an example of a sample valid [but unrealistic] bidding sequence. North is the dealer and East/West end up winning the contract of 4♥ doubled with West as the declarer.

North	East	South	West
1♣	1♦	1♥	double
1♠	1NT	double	redouble
2♦	pass	pass	double
pass	pass	redouble	2♥
pass	4♥	pass	pass
double	pass	pass	pass

You will complete the following partially defined class and functions

```

1  from a10q1 import *
2  import check
3
4
5  class Bid:
6      '''
7      Fields:
8          value (Str)
9          suit (anyof Str None)

```

```

10     Requires:
11         value is one of "1", "2", "3", "4", "5", "6", "7",
12         "pass", "double", "redouble"
13         suit is one of "C", "D", "H", "S", "NT" or None
14         If value is non-numeric, then suit must be None
15     '''
16
17     def __init__(self, bid_value, bid_suit):
18         '''
19         Initialized a valid Bridge bid.
20
21         Effects: Mutates self
22
23         __init__: Bid Str (anyof Str None) -> None
24         Requires: Conditions from Fields above are met.
25         '''
26         self.value = bid_value
27         self.suit = bid_suit
28
29     def __repr__(self):
30         '''
31         Returns a representation of a Card object
32
33         __repr__: Card -> Str
34         '''
35         if self.suit == None:
36             return "{0.value}".format(self)
37         return "{0.value}{0.suit}".format(self)
38
39     def __eq__(self, other):
40         '''
41         Returns True if self and other have equal values and suits
42         and False otherwise
43
44         __eq__: Bid Any -> Bool
45         '''
46         ##YOUR CODE GOES HERE
47         pass
48
49     def __lt__(self, other):
50         '''

```

A brief summary of the functions and methods you need to complete is below:

a. Submit and complete the magic method for the Bid class

\_\_eq\_\_(self, other)

that returns `True` if both values are `Bids` and the suits and values of the two `Bids` are equal and `False` otherwise.

b. Code the magic method for the `Bid` class

```
__lt__(self, other)
```

that returns `True` if both values are `Bids`, the `Bids` are numeric and the first `Bid` occurs before the second `Bid` and `False` otherwise. Note that `__lt__` overrides `<` much like `__eq__` overrides `==`.

c. Write the code for the function

```
valid_bid(bids, new_bid)
```

that consumes a list of `Bids` and another `Bid` and returns `True` if this bid is legal according to the rules above and `False` otherwise. Note this function assumes every bid that has been made thus far has been legal.

d. Write the code for the function

```
bidding_complete(bids)
```

that consumes a valid bidding list of `Bids` and returns `True` if a contract has been established, that is, the bidding has ended with an all-pass contract or with some non-pass bid followed by three consecutive passes, and returns `False` otherwise.

e. Submit and complete the function


```
contract(bids)
```

that consumes a list of `Bids` satisfying `bidding_complete(bids) == True` and returns the contract that was settled on including an all passed contract, and/or any doubling or redoubling that occurred.

For testing purposes, we have also provided you with the `declarer` code and a `bidding_bootstrap` function which consumes an optional `deck` parameter you can use in conjunction with the `deck_bootstrap` code from the previous problem to deal cards and simulate a bidding sequence in Bridge. The function returns a list of players (following our convention from the previous part that the dealer is last) and a list of the bids so that you could use this to determine the declarer and the contract for the next part. Note this will only work once you've completed the parts above.

**Note:** The command `from a10q1 import *` allows you to use our version of the functions from a10q1. Any helper functions you created in the previous problem however would need to be copied over here if you wanted to use them. The functions you were given from the previous problem though can be used in this problem as though they were given in this question.

Last submission on 04-08-2021 00:49:57 Result: Could not find test results

main.py  - Add File

```

1  from a10q1 import *
2  import check
3
4
5  class Bid:
6      '''
7      Fields:
8          value (Str)
9          suit (anyof Str None)
10     Requires:
11         value is one of "1", "2", "3", "4", "5", "6", "7",
12         "pass", "double", "redouble"
13         suit is one of "C", "D", "H", "S", "NT" or None
14         If value is non-numeric, then suit must be None
15     '''
16
17     def __init__(self, bid_value, bid_suit):
18         '''
19         Initialized a valid Bridge bid.
20
21         Effects: Mutates self
22
23         __init__: Bid Str (anyof Str None) -> None
24         Requires: Conditions from Fields above are met.
25         '''
26         self.value = bid_value
27         self.suit = bid_suit
28
29     def __repr__(self):
30         '''
31         Returns a representation of a Card object
32
33         __repr__: Card -> Str
34         '''
35         if self.suit == None:
36             return "{0.value}".format(self)
37         return "{0.value}{0.suit}".format(self)
38

```

```
38 ~
39 def __eq__(self, other):
40     '''
41     Returns True if self and other have equal values and suits
42     and False otherwise
43
44     __eq__: Bid Any -> Bool
45     '''
46     ##YOUR CODE GOES HERE
47     pass
48
49 def __lt__(self, other):
50     '''
```

Code Output

Run Code

Save Code

Reset Code

Show History

Visualize

Submit Code

View Submissions

Discussion

Hide Discussion

Topic: Module\_10 / Final Project: Bridge - Bidding

Add a Post

Show all posts

by recent activity

There are no posts in this topic yet.