Invocation of the program

./bvDHT.py <peer IP address> <peer port>

./bvDHT.py (no params if starting new DHT)

Things to implement

- Basic Hash Table Functions
  - Get [DONE]
  - Insert (also updates) [DONE]
  - Remove [DONE]
  - Contains [DONE]
- Common Functions
  - Locate [DONE]
- Connectivity
  - Connect [DONE]
  - Disconnect [DONE]
  - Update Prev (on next node) [DONE]

Primitives

- Command Messages: (All Caps) \n
- PeerAddress: "IP:port" \n
- Hashed Key: str(160-bit integer) \n
- Acknowledgement: Yes (1\n) or No (0\n)
- Integers in general: str (int) \n

Locate

- [Self->Peer] LOCATE
- [Self->Peer] HashedKey
- [Peer->Self] PeerAddress

Connect

- [Self->Peer] CONNECT
- [Self->Peer] HashedKey (of Self's PeerAddress)
- [Peer->Self] Acknowledgement
  if 1, continue on – if 0, bail out of protocol
- Transfer all entries
  - [Peer->Self] integer numEntries
  - For loop – numEntries times do the following:
    - [Peer->Self] HashKey of entry
    - [Peer->Self] integer len(ValueData)
    - [Peer->Self] byteArray of ValueData
- [Peer->Self] PeerAddress of it's Next peer
- Complete Update Prev on Next Node sub-protocol
- [Self->Peer] PeerAddress of Self
- *** Ownership Officially Transferred by completing this ***

Disconnect

- [Self->Prev] DISCONNECT
- [Self->Prev] Self's Next PeerAddress
- Transfer all entries
  - [Self->Prev] integer numEntries
  - For loop – numEntries times do the following:
    - [Self->Prev] HashKey of entry
    - [Self->Prev] integer len(ValueData)
    - [Self->Prev] byteArray of ValueData
- Prev performs UpdatePrev on Next
- [Prev->Self] Acknowledgement
  - *** Ownership Officially Transferred by completing this ***

Update Prev

- [Self->Next] UPDATE_PREV
- [Self->Next] PeerAddress of self
- [Next->Self] Acknowledgement

Contains

- [Self->Peer] CONTAINS
- [Self->Peer] HashedKey
- [Peer->Self] Acknowledgement of ownership of HashedKey Space
  Bail out if answer is ‘0’\n
- [Peer->Self] Acknowledgement of having entry

Get

- [Self->Peer] GET
- [Self->Peer] HashedKey
- [Peer->Self] Acknowledgement of ownership of HashedKey Space
  Bail out if answer is ‘0’\n
- [Peer->Self] integer len(ValueData)
- [Peer->Self] byteArray of ValueData

Insert

- [Self->Peer] INSERT
- [Self->Peer] HashedKey
- [Peer->Self] Acknowledgement of ownership of HashedKey Space
  Bail out if answer is ‘0’\n
- [Self->Peer] integer len(ValueData)
- [Self->Peer] byteArray of ValueData
- [Peer->Self] Acknowledgement of successful INSERT

Remove

- [Self->Peer] REMOVE
- [Self->Peer] HashedKey
- [Peer->Self] Acknowledgement of ownership of HashedKey Space
  Bail out if answer is ‘0’\n
- [Peer->Self] Acknowledgement of successful REMOVE
  - Also acknowledge ‘1’ if key didn’t exist. Remove didn’t fail.