CB U3.10 & U 3.11: Lists and Binary Search

Lists

Lists are a collection of data of the same type. Lists are traversed by iteration and loops. Common list algorithims include determining the min or max of a list, and the sum and mean.

List algorithims consist of iteration and operations on each individual element:

```
keep_track = 0
for x in list:
    #... do operation on x, keep track by modifying another variable

#... do processing on keep_track (or not)
return keep_track
```

Indexing

Linear and sequential search algorithims check each element in a list until the target value is found or the list has been exhausted. Lists are indexed starting from 1 until length of the list.

```
n = ['a','b','c']
n[1] = 'a'
n[2] = 'b'
n[3] = 'c'
```

Search on list

```
for n in list:
    if n==target:
        # exit or etc.
```

Binary Search

Binary search starts at the middle of sorted set of data and eliminates half of the data repeatedly until the target value is found

Data must be sorted.

Binary search is faster than linear search when sorting the list is negligible.

A list of length 2^n will take n operations to sort.