Prolegomena unit outline:

- Algorithms and correctness (last week Friday and today)
- Algorithms and efficiency (today and the rest of the week)
- Abstract data types (next week Wednesday)
- Data Structures (next week Friday and the Monday after)

Today:

- Finish check-sorting problem
- “Binary search” problem
- Class invariants (LinkedList)
- Start efficiency
What good are invariants?

▶ They are a tool for reasoning about the state and progress of an algorithmic process.
▶ They are a way to explain the meaning of a variable and capture how the variables relate to each other.
▶ They help with testing and debugging.
▶ They are a means for proving that an algorithm is correct.
Invariant (Class LinkedList)

(a) $\text{head} = \text{null} \iff \text{tail} = \text{null} \iff \text{size} = 0$.

(b) If $\text{tail} \neq \text{null}$ then $\text{tail}.\text{next} = \text{null}$.

(c) If $\text{head} \neq \text{null}$ then $\text{tail}$ is reached by following $\text{size} - 1$ next links from $\text{head}$. 
```python
def bounded_linear_search(sequence, P):
    found = False
    i = 0
    while not found and i < len(sequence):
        found = P(sequence[i])
        i += 1
    if found:
        return i - 1
    else:
        return -1
```

\[
T_{bls}(n) = a_0 + a_1(n+1) + a_2n + a_3 + \max(a_4, a_5)
= b_0 + b_1n
\]
def binary_search(sequence, T0, item):
    low = 0
    high = len(sequence)
    while high - low > 1:  # $c_1 \lg n + 1$
        mid = (low + high) / 2
        compar = T0(item, sequence[mid])
        if compar < 0:  # item comes before mid
            high = mid
        elif compar > 0:  # item comes after mid
            low = mid + 1
        else:  # item is at mid
            assert compar == 0
            low = mid
            high = mid + 1
    if low < high and T0(item, sequence[low]) == 0:  # $c_3$
        return low
    else:
        return -1

$T_{bs}(n) = c_0 + c_1 (\lg n + 1) + c_2 \lg n + c_3 + \max(c_4, c_5)$
$          = d_0 + d_1 \lg n$
\[
T_{sel}(n) = f_1 + f_2 n + f_3 n^2
\]
Coming up:

Due Wednesday, Aug 31 (end of day):
Read Section 2.1
Do Ex 1.(6 & 7) (shoot for class time)
Take quiz

Due Friday, Sept 2 (end of day):
Read Sections 1.(3 & 4) (spread out)
Do practice problems 1.(27 & 28) and 1.(42 & 43)
Take quiz