Invariant (Outer loop of selection_sort)

- (a) The range [0, i) in sequence is sorted.
- (b) All the elements in range [0, i) in sequence are less than or equal to all the elements in the range [i, n).
- (c) $0 \leq i \leq n$.

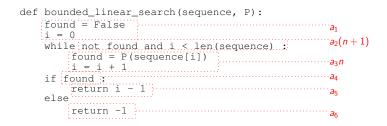
Invariant (Inner loop of selection_sort)

(b) min is the smallest element in the range [i,j). (Formally: ∀ k ∈ [i,j), min ≤ sequence[k].)

(c) $i \leq \min_{j \in \mathbb{N}} os < j \leq n$.

Correctness claim (selection_sort.)

After n iterations, sequence is sorted and selection_sort returns.



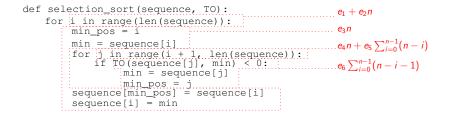
$$T_{bls}(n) = a_1 + a_2(n+1) + a_3n + a_4 + \max(a_5, a_6) \\ = b_0 + b_1n$$

◆□ → ◆□ → ◆三 → ◆三 → ○へ ⊙

<pre>def binary_search(sequence, TO, item):</pre>
low = 0
high = len(sequence)
while high - low > 1 :
mid = (low + high) / 2
<pre>compar = TO(item, sequence[mid])</pre>
if compare < 0:
high = mid
elif compar > 0: C5
low = mid + 1
else:
low = mid
high = mid + 1
if low < high and TO(item, sequence[low]) == 0:
return low
else:
return -1

$$T_{bs}(n) = c_1 + c_2(\lg n + 1) + (c_3 + \max(c_4, c_5 + c_6, c_5 + c_7)) \lg n + c_8 + \max(c_9, c_{10}) = d_0 + d_1 \lg n$$

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 のへで



$$T_{sel}(n) = f_1 + f_2 n + f_3 n^2$$

◆□▶ ◆□▶ ◆□▶ ◆□▶ ◆□ ● のへの

Every time you run a program, you are performing a scientific experiment that relates the program to the natural world and answers one of our core questions: How long will my program take? The running time [of many programs] is relatively insensitive to the input itself; it depends primarily on the problem size.

Sedgewick, pg 173

 $g(n) \sim f(n)$ means the functions are asymptotically equal, that is, that $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 1$. Thus $\frac{n^3}{6} - \frac{n^2}{2} + \frac{n}{3} \sim \frac{n^3}{6}$.

g(n) = O(f(n)), which really should be written $g(n) \in O(f(n))$, means that a scaled version of f(n) asymptotically bounds gabove. It means there exists a c such that when n is large enough, $g(n) \leq cf(n)$. Thus $\frac{n^3}{6} - \frac{n^2}{2} + \frac{n}{3} = O(\frac{n^3}{6})$ but also $\frac{n^3}{6} - \frac{n^2}{2} + \frac{n}{3} = O(n^3)$ and $\frac{n^3}{6} - \frac{n^2}{2} + \frac{n}{3} = O(n^4)$.

With big-oh, you can throw away the lower ordered terms *and* throw away the constant factor of the highest order term *and* overshoot.

With tilde, you only can throw away the lower ordered terms.