```
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
```

a_0

$a_1(n+1)$

$a_2 n$

a_3

a_4

a_5

$$
\begin{aligned}
T_{bls}(n) &= a_1 + a_2(n+1) + a_3 n + a_4 + \max(a_5, a_6) \\
&= b_0 + b_1 n
\end{aligned}
$$

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

c_0

$c_1(\lg n + 1)$

$c_2 \lg n$

$c_3 \lg n$

$c_5 \lg n$

$\max(c_4, c_6, c_7) \lg n$

c_8

$\max(c_9, c_{10})$

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

```python
def selection_sort(sequence, T0):
    for i in range(len(sequence)) :
        min_pos = i
        min = sequence[i]
        for j in range(i + 1, len(sequence)):
            if T0(sequence[j], min) < 0 :
                min = sequence[j]
                min_pos = j
        sequence[min_pos] = sequence[i]
        sequence[i] = min
```

$e_0 + e_1 n$

$e_2 n$

$e_3 n + e_4 \sum_{i=0}^{n-1} (n - i - 1)$

$e_5 \sum_{i=0}^{n-1} (n - i - 1)$

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