CSC 236 Tutorial 8

Harry Sha

July 12, 2023

Selection Sort

Descending Sequences

Selection Sort

```python
def selection_sort(l):
    n = len(l)
    for i in range(n):
        min_idx = i
        for j in range(i+1, n):
            if l[j] < l[min_idx]:
                min_idx = j
        l[i], l[min_idx] = l[min_idx], l[i]
    return l
```

Trace

To get an idea of how this algorithm works, trace the algorithm on input $I = [3, 236, 36, 23, 2, 6]$

Notation

If l_1 and l_2 are lists, then define $l_1 \preceq l_2$ to mean $\forall x \in l_1, \forall b \in l_2 . a \leq b$. That is everything in l_1 is less than anything in l_2.
If l_1 is a number, then $l_1 \preceq l_2$ is short for $[l_1] \preceq l_2$, same for l_2.

Lemma

What do lines 4-8 do? If it helps, you can imagine lines 4-8 being extracted as a helper function that takes in i and l.

What are the preconditions and postconditions of lines 4-8?

Make a claim about these lines and prove it is correct!

Lemma

Lemma. If l is a list of natural numbers, and $i \in \mathbb{N}$, such that $0 \leq i < \text{len}(l)$ at the start of line 4, by line 8, $l[i]$ is swapped with the $l[k]$ where $l[k] \preceq l[i+1:]$.

To prove the lemma, it suffices to show that at the beginning of line 8, `min_idx` is such that $l[\texttt{min\_idx}]$ is minimal in $l[i+1:]$ (since line 8 does the swapping).

We now analyze lines 4-7.

Proof of Lemma

Precondition. l is a list of natural number of length n, $i \in \mathbb{N}$, such that $0 \leq i < n$. **Postcondition.** $l[\texttt{min\_idx}] \preceq l[i+1:]$

Proof of Lemma

Loop Invariant. $P(k)$. After the kth iteration,

a.) $j = i + 1 + k$.

b.) $l[\texttt{min\_idx}] \preceq l[i + 1 : j]$.

Proof of Lemma

Initialization. $P(0)$. j is initialized to $i + 1$, and $l[i + 1 : j] = l[i + 1 : i + 1] = []$, and $l[\texttt{min\_idx}] \preceq []$ is vacuously true.

Proof of Lemma

Maintenance. Let $k \in \mathbb{N}$ and suppose $P(k)$. We'll show $P(k+1)$. Note that j is incremented by the for loop and hence $j_{k+1} = j_k + 1 = i + 1 + k + 1$, so $P(k+1)$.a is true. We'll now show $P(k+1)$.b.

By the inductive hypothesis, $l[\texttt{min\_idx}_k] \preceq l[i+1 : j_k]$. If $l[j_k] < l[\texttt{min\_idx}]$, then since $\texttt{min\_idx}_{k+1} = j$, and otherwise $\texttt{min\_idx}_{k+1} = \texttt{min\_idx}_k$. In the first case, $l[j_k] < l[\texttt{min\_idx}]$, and $l[\texttt{min\_idx}] \preceq l[i+1 : j_k]$, so $l[j_k] \preceq l[i+1 : j_k+1]$. In the second case $l[j_k] \geq l[\texttt{min\_idx}]$, so $l[\texttt{min\_idx}] \preceq l[i+1 : j_k+1]$, since $j_{k+1} = j_k + 1$, this completes the induction.

Proof of Lemma

Termination. The for loop terminates after iteration k where $j_k = n$. By $P(k)$.b, we have that $l[\texttt{min\_idx}_k] \preceq l[i+1 : j_k]$. Since $j_k = n$, we have $l[\texttt{min\_idx}_k] \preceq l[i+1 :]$
This concludes the proof of the claim.

Correctness

Armed with this lemma, prove the correctness of `selection_sort`.

Solutions

Precondition. l is a list of of natural numbers.
Postcondition. Returns a sorted version of l

Solutions

Loop Invariant. $P(k)$: After iteration k,

a.) $i = k$.

b.) $l[: k]$ is sorted

c.) $l[: k] \preceq l[k :]$

Solutions

Initialization. $P(0)$:

a.) i is initialized to 0.

b.) $l[:0] = []$ is (vacuously) sorted.

c.) Since $l[:0] = []$, $l[:k] \preceq l[k:]$ is also vacuously true.

Solutions

Maintenance. Let $k \in \mathbb{N}$, and suppose $P(k)$. We'll show $P(k+1)$.

$P(k+1)$.a follows from the mechanics of the for loop (it just gets incremented by 1).

By $P(k)$.b, we have $l_k[: k]$ is sorted. By the lemma, we have that l_{k+1} is l_k with a minimal element in $l_k[k :]$ swapped to index k. By $P(k)$.c, we have $l_k[: k] \preceq l_k[k :]$, so $l_{k+1}[: k] \preceq l_{k+1}[k]$. Thus, $l_{k+1}[: k+1]$ is still sorted, so $P(k+1)$.b holds.

Then, since $l_{k+1}[k] \preceq l_k[k :]$, and $l_k[: k] \preceq l_k[k :]$, we have $l_{k+1}[: k+1] \preceq l_{k+1}[k+1 :]$, so $P(k+1)$.c holds.

Solutions

Termination. The for loop terminates after n iterations. By $P(n)$.b, we have $l_n[: n] = l$ is sorted, as required.

Selection Sort

Descending Sequences

Descending Sequences

Prove the following loops terminate. The following examples come from Course Notes by Vassos Hadzilacos.

Hint - for some of them it might be easier to use the descending sequence method.

1

```
▶ Precondition: x, y ∈ ℕ.
▶ Postcondition: True.
while x ≠ 0 or y ≠ 0 do
    if x ≠ 0 then
        x := x − 1
    else
        x := 16
        y := y − 1
    end if
end while
```

Figure 2.3: A loop with an interesting proof of termination

Solution (sketch)

$x + 17y$. First, need to show that $x + 17y$ is always a natural number (i.e. it doesn't go negative at some point). Prove this formally by induction. Informal argument: x and y at decremented by at most 1 in each iteration. If the while check passes, x, y are both at least 1. Thus, x, y are always non-negative.

The next step is to show that $x + 17y$ is decreasing.

Let x_n, y_n be the value of x and y at the start of the nth iteration. Show $x_{n+1} + 17y_{n+1} < x_n + 17y_n$ in each case split (exercise).

2

▶ Precondition: $x, y \in \mathbb{N}$ and x is even.

while $x \neq 0$ **do**
    if $y \geq 1$ **then**
        $y := y - 3;\ x := x + 2$
    else
        $x := x - 2$
    end if
end while

Solution (sketch)

$2 + y + x$. Show this is always non-negative. Sketch: Show $y \geq -2$, and $x \geq 0$. For the $x \geq 0$ part, use the fact that x starts off as even and is only ever incremented/decremented by 2. Again, the formal proof is by induction.

To show that this is decreasing, note that the value of $2 + y + x$ either decreases by 1 in the `if` case or decreases by 2 in the `else` case.

9. Prove that the following program halts for every input $x \in \mathbb{N}$.

```
▶ Precondition: x ∈ ℕ.
    y := x * x
    while y ≠ 0 do
        x := x − 1
        y := y − 2 * x − 1
    end while
```

Hint: Derive (and prove) a loop invariant whose purpose is to help prove termination.

Solution

Let $P(n)$ be the predicate. $y_n = x_n^2$ and $x_n = x_0 - n$. We'll show that for all $n \in \mathbb{N}.(P(n))$.

Base case. $y_0 = x_0 \cdot x_0 = x_0^2$. Also $x_0 = x_0 - 0$

Inductive step. Let $k \in \mathbb{N}$ and suppose $P(k)$. Then, we have

$$\begin{aligned} y_{k+1} &= y_k - 2x_{k+1} - 1 \\ &= x_k^2 - 2(x_k - 1) - 1 \\ &= x_k^2 - 2x_k + 1 \\ &= (x_k - 1)^2 \\ &= x_{k+1}^2 \end{aligned}$$

Thus, at the start of iteration x_0, $y_{x_0} = (x_{x_0})^2 = (x_0 - x_0)^2 = 0$, thus the while check fails and the loop terminates.