Homework 9: Core solutions

Section 8.2 on page 264 problems 13b, 27a-27b.
Section 8.3 on page 275 problems 1b, 8, 10a-10b, 14.
Section 8.4 on page 279 problems 2c, 7a.
Chapter 8 review problems on page 280 problem 6.

13b. Show that \( b^n < n! \) for any \( b > 1 \)

**Solution:** Let \( a_n = \frac{b^n}{n!} \). Then \( \frac{a_{n+1}}{a_n} = \frac{b^{n+1}/(n+1)!}{b^n/n!} = \frac{b}{n+1} \), which converges to zero as \( n \) tends to infinity. By the ratio test (from Calculus), the series \( \sum_{n=1}^{\infty} a_n \) converges absolutely. A necessary condition for the series to converge, however, is that the terms \( a_n \) must tend to zero. Hence, \( \lim_{n \to \infty} \frac{b^n}{n!} = 0 \). By Proposition 8.2.6 on page 256, we have \( b^n \prec n! \).

27a. Establish the triangle inequality:

\[ |a + b| \leq |a| + |b|. \]

**Solution:** For any real number \( x \) and \( y \) we have

\[ -|x| \leq x \leq |x|, \quad \text{and} \]
\[ -|y| \leq y \leq |y|. \]

Adding the two inequalities together we have

\[ -(|x| + |y|) \leq x + y \leq |x| + |y|. \] (1)

But

\[ |x + y| = \begin{cases} 
  x + y & \text{if } x + y \geq 0 \\
  -(x + y) & \text{if } x + y < 0.
\end{cases} \]

In either case, \( |x + y| \leq |x| + |y| \) using Equation (1).

**Alternate proof:** We first show that \( |x + y|^2 \leq (|x| + |y|)^2 \). We have

\[ |x + y|^2 = (x + y)(x + y) = x^2 + 2xy + y^2 = |x|^2 + 2xy + |y|^2 \leq |x|^2 + 2|x||y| + |y|^2 = (|x| + |y|)^2. \]

Now, if \( a, b \) are any positive real numbers and \( a \leq b \), then \( \sqrt{a} \leq \sqrt{b} \). This is immediate since the function \( f(x) = \sqrt{x} \) is strictly increasing on it’s domain (recall, \( f’(x) = \frac{1}{2\sqrt{x}} > 0 \) for \( x > 0 \)). This finishes the proof.
27b. Show that $|x_1 + \cdots + x_n| \leq |x_1| + \cdots + |x_n|$ for any $n \geq 1$ and real numbers $x_1, \ldots, x_n$. Solution: Proof by induction. When $n = 1$ we have $|x_1| \leq |x_1|$ (by the way, the $n = 2$ case is just what we did in the previous exercise). Now, suppose $|x_1 + \cdots + x_k| \leq |x_1| + \cdots + |x_k|$. We have,

$$|x_1 + \cdots + x_k + x_{k+1}| \leq |x_1 + \cdots + x_k| + |x_{k+1}|$$

$$\leq |x_1| + \cdots + |x_k| + |x_{k+1}|,$$

where the first inequality follows from the previous exercise and the second equality follows from the induction hypothesis.

1b. Show the sequence of steps in a binary search to find $x = 7$ in the list $1, 2, 3, 4, 5, 6, 7, 8, 9$. How many times is $x$ compared with an element in the list? How many times would it be compared if we used a linear search? Solution: Recall the binary search and linear search algorithms.

**Binary search algorithm:**

**Input:** $a_1, \ldots, a_n, x$ with $a_1 \leq a_2 \leq \cdots \leq a_n$.

**Procedure:**

**STEP 1:** Initialize $S = 0$.

**STEP 2:** WHILE $n > 0$,

IF $n = 1$ then

IF $x = a_1$ set $n = 0$ and replace $S$ with 1.

ELSE set $n = 0$.

ELSE

set $m = \lfloor \frac{n}{2} \rfloor$;

IF $x \leq a_m$ replace the current list with $a_1, \ldots, a_m$ and set $n = m$;

ELSE replace the current list with $a_{m+1}, \ldots, a_n$ and replace $n$ by $n - m$.

END WHILE

**Output:** $S$.

**Linear search algorithm:**

**Input:** $a_1, \ldots, a_n, x$.

**Procedure:**

**STEP 1:** Initialize $S = 0$.

For $i = 1..n$, 
IF $x = a_i$, set $i = 2n$ and replace $S$ with 1.

**Output:** $S$.

The algorithms above each take a list and output 1 if $x$ is an element of the list and output 0 otherwise.

The steps in the binary search with the list 1, 2, 3, 4, 5, 6, 7, 8, 9 and $x = 7$ is as follows:

Initialize $S = 0$. Since $n \neq 1$ set $m = \left\lfloor \frac{9}{2} \right\rfloor = 4$. The number $x = 7$ does not satisfy $x \leq a_m$ since $7 \nleq a_4 = 4$. So we replace the list by 5, 6, 7, 8, 9. Since $n \neq 1$ set $m = \left\lfloor \frac{5}{2} \right\rfloor = 2$. The number $x = 7$ does not satisfy $x \leq a_m$ since $a_2 = 6$, so we replace the list by 7, 8, 9. Since $n \neq 1$ set $m = \left\lfloor \frac{3}{2} \right\rfloor = 1$. The number $x = 7$ does satisfy $x \leq a_1$ since $a_1 = 7$. Hence, we replace the list with the list “7”. Now, $n = 1$ and $x = a_1$, so we replace $S$ with 1. Finally, we output $S = 1$.

There are a total of 4 comparisons using the binary search.

If we were to do a linear search, there would be 7 comparisons (6 failed comparison and then a successful comparison).

\[\square\]

8. Show the sequence of steps involved in merging the sorted lists 2, 4, 4, 6, 8 and 1, 5, 7, 9, 10. How many comparisons are required? **Solution:** Recall the merging algorithm.

**Merging algorithm:** $\text{MERGE}(L_1, L_2)$

**Input:** $L_1 = (a_1, \ldots, a_s)$ and $L_2 = (b_1, \ldots, b_t)$ with $a_1 \leq a_2 \leq \cdots \leq a_s$ and $b_1 \leq b_2 \leq \cdots \leq b_t$.

**Procedure:**

**STEP 1:** Initialize $L_3 = ()$.

**STEP 2:**

IF $L_1$ is empty, set $L_3 = L_2$ and STOP.

IF $L_2$ is empty, set $L_3 = L_1$ and STOP.

**STEP 3:**

IF $a_1 \leq b_1$, remove $a_1$ from $L_1$ and append it to $L_3$; if this empties $L_1$ then append $L_2$ to $L_3$ and STOP. Relabel the elements in $L_1$ and repeat Step 3.

ELSE $a_1 > b_1$, remove $b_1$ from $L_2$ and append it to $L_3$; if this empties $L_2$ then append $L_1$ to $L_3$ and STOP. Relabel the elements in $L_2$ and repeat Step 3.

**Output:** $L_3$. 
The sequence of steps is as follows:

<table>
<thead>
<tr>
<th></th>
<th>(\mathcal{L}_1)</th>
<th>(\mathcal{L}_2)</th>
<th>(\mathcal{L}_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2, 4, 4, 6, 8)</td>
<td>(1, 5, 7, 9, 10)</td>
<td>()</td>
</tr>
<tr>
<td>2</td>
<td>(2, 4, 4, 6, 8)</td>
<td>(5, 7, 9, 10)</td>
<td>(1)</td>
</tr>
<tr>
<td>3</td>
<td>(4, 4, 6, 8)</td>
<td>(5, 7, 9, 10)</td>
<td>(1, 2)</td>
</tr>
<tr>
<td>4</td>
<td>(4, 6, 8)</td>
<td>(5, 7, 9, 10)</td>
<td>(1, 2, 4)</td>
</tr>
<tr>
<td>5</td>
<td>(6, 8)</td>
<td>(5, 7, 9, 10)</td>
<td>(1, 2, 4, 4)</td>
</tr>
<tr>
<td>6</td>
<td>(6, 8)</td>
<td>(7, 9, 10)</td>
<td>(1, 2, 4, 4, 5)</td>
</tr>
<tr>
<td>7</td>
<td>(8)</td>
<td>(7, 9, 10)</td>
<td>(1, 2, 4, 4, 5, 6)</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>(9, 10)</td>
<td>(1, 2, 4, 4, 5, 6, 7)</td>
</tr>
<tr>
<td>9</td>
<td>()</td>
<td>(9, 10)</td>
<td>(1, 2, 4, 4, 5, 6, 7, 8)</td>
</tr>
<tr>
<td>10</td>
<td>()</td>
<td>()</td>
<td>(1, 2, 4, 4, 5, 6, 7, 8, 9, 10)</td>
</tr>
</tbody>
</table>

There are 8 comparisons required (the 1st and 10th step did not require a comparison).

10. Find an example of two ordered lists of lengths \(s\) and \(t \geq 3\) that can be merged with

(a) one comparison. Solution: Set \(\mathcal{L}_1 = (1)\) and \(\mathcal{L}_2 = (2, 3, 4)\).

(b) \(t\) comparisons. Solution: Set \(\mathcal{L}_\infty = (t + 1)\) and \(\mathcal{L}_2 = (1, 2, 3, \ldots, t)\).

11. Sort the list 7, 2, 2, 5, 3, 5, 4 using bubble sort and merge sort. In each case, how many comparisons were needed? (for merge sort, you may ignore comparisons required to check the size and parity of \(n\) at each iteration of Step 3) Solution: The merge sort and bubble sort algorithms are stated below. Recall the **Merging algorithm** above whose input \(\mathcal{L}_1, \mathcal{L}_2\) are two ordered lists and whose output \(\text{MERGE}(\mathcal{L}_1, \mathcal{L}_2)\) is the merged ordered list coming from \(\mathcal{L}_1, \mathcal{L}_2\).

**Merge sort algorithm:**

**Input:** An unordered list \(\mathcal{L} = (a_1, \ldots, a_n)\).

**Procedure:**

1. **STEP 1:** Initialize \(F = 0\).
2. **STEP 2:** For \(i = 1..n\),
   - define \(\mathcal{L}_i\) to be the list with single element \(a_i\).
3. **STEP 3:** WHILE \(F = 0\),
   - IF \(n = 1\), set \(F = 1\).
   - IF \(n = 2m\) is even,
For $i = 1..m$
    replace $\mathcal{L}_i$ with $\text{MERGE}(\mathcal{L}_{2i}, \mathcal{L}_{2i-1})$.
Set $n := m$.
IF $n = 2m + 1$ is odd and $n \neq 1$,
    For $i = 1..m$
        replace $\mathcal{L}_i$ with $\text{MERGE}(\mathcal{L}_{2i}, \mathcal{L}_{2i-1})$.
    Set $\mathcal{L}_{m+1} := \mathcal{L}_n$.
    Set $n := m + 1$.
Output: $\mathcal{L}_1$.

Bubble sort algorithm:
Input: An unordered list $\mathcal{L} = (a_1, \ldots, a_n)$.
Procedure:
STEP 1:
For $i = n - 1$ down to 1,
    For $j = 1..i$,
        IF $a_j > a_{j+1}$, set $a_j := a_{j+1}$ and $a_{j+1} := a_j$.
Output: $\mathcal{L}$. 