

# 4005-800 ALGORITHMS

## HOMEWORK 4

Christopher Wood

April 25, 2012

**PROBLEM 1-a.** *Prove that for any  $n \in \mathbb{N}$ ,  $\lfloor (n+1)/2 \rfloor = \lceil n/2 \rceil$ .*

**Solution.**

**Case 1:**  $n$  is even ( $2 \mid n$ , or  $n = 2m$  for some  $m \in \mathbb{N}$ )

$$\begin{aligned} \left\lfloor \frac{n+1}{2} \right\rfloor &= \left\lfloor \frac{2m+1}{2} \right\rfloor \text{ (by substitution)} \\ &= \left\lfloor \frac{2m}{2} + \frac{1}{2} \right\rfloor \text{ (by partial fractions)} \\ &= \left\lfloor m + \frac{1}{2} \right\rfloor \text{ (by division)} \\ &= m + \left\lfloor \frac{1}{2} \right\rfloor \text{ (by property of floor function)} \\ &= m \text{ (by property of floor function)} \\ &= \frac{2m}{2} \text{ (by division)} \\ &= \left\lceil \frac{2m}{2} \right\rceil \text{ (by property of ceiling function)} \\ &= \left\lceil \frac{n}{2} \right\rceil \text{ (by substitution)} \end{aligned}$$

**Case 2:**  $n$  is odd ( $2 \nmid n$ , or  $n = 2m+1$  for some  $m \in \mathbb{N}$ )

$$\begin{aligned} \left\lfloor \frac{n+1}{2} \right\rfloor &= \left\lfloor \frac{(2m+1)+1}{2} \right\rfloor \text{ (by substitution)} \\ &= \left\lfloor \frac{2(m+1)}{2} \right\rfloor \text{ (by factoring)} \\ &= m+1 \text{ (by division and property of floor function)} \\ &= m + \left\lceil \frac{1}{2} \right\rceil \text{ (by property of ceiling function)} \\ &= \frac{2m}{2} + \left\lceil \frac{1}{2} \right\rceil \text{ (by division)} \\ &= \left\lceil \frac{2m}{2} + \frac{1}{2} \right\rceil \text{ (by property of ceiling function)} \\ &= \left\lceil \frac{2m+1}{2} \right\rceil \text{ (by partial fractions)} \\ &= \left\lceil \frac{n}{2} \right\rceil \text{ (by substitution)} \end{aligned}$$

Thus, since a number  $n \in \mathbb{N}$  can only be even or odd, we can conclude that for any  $n \in \mathbb{N}$ ,  $\left\lfloor \frac{n+1}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil$ .

**PROBLEM 1-b.** Prove that for any  $n \in \mathbb{N}$ ,  $\lfloor n/2 \rfloor + 1 = \lceil (n+1)/2 \rceil$ .

**Solution. Case 1:**  $n$  is even ( $2 \mid n$ , or  $n = 2m$  for some  $m \in \mathbb{N}$ )

$$\begin{aligned}
\left\lfloor \frac{n}{2} \right\rfloor + 1 &= \left\lfloor \frac{2m}{2} \right\rfloor + 1 \text{ (by substitution)} \\
&= \frac{2m}{2} + 1 \text{ (by property of floor function)} \\
&= \frac{2m}{2} + \left\lceil \frac{1}{2} \right\rceil \text{ (by property of ceiling function)} \\
&= \left\lceil \frac{2m}{2} \right\rceil + \left\lceil \frac{1}{2} \right\rceil \text{ (by property of ceiling function)} \\
&= \left\lceil \frac{2m}{2} + \frac{1}{2} \right\rceil \text{ (by property of ceiling function)} \\
&= \left\lceil \frac{2m+1}{2} \right\rceil \text{ (by partial fractions)} \\
&= \left\lceil \frac{n+1}{2} \right\rceil \text{ (by substitution)}
\end{aligned}$$

**Case 2:**  $n$  is odd ( $2 \nmid n$ , or  $n = 2m + 1$  for some  $m \in \mathbb{N}$ )

$$\begin{aligned}
\left\lfloor \frac{n}{2} \right\rfloor + 1 &= \left\lfloor \frac{2m+1}{2} \right\rfloor + 1 \text{ (by substitution)} \\
&= \left\lfloor \frac{2m}{2} + \frac{1}{2} \right\rfloor + 1 \text{ (by partial fractions)} \\
&= m + \left\lfloor \frac{1}{2} \right\rfloor + 1 \text{ (by division and property of floor function)} \\
&= m + 1 \text{ (by property of floor function)} \\
&= \frac{2(m+1)}{2} \text{ (by division)} \\
&= \left\lceil \frac{2(m+1)}{2} \right\rceil \text{ (by property of ceiling function)} \\
&= \left\lceil \frac{2m+2}{2} \right\rceil \text{ (by distribution)} \\
&= \left\lceil \frac{(2m+1)+1}{2} \right\rceil \text{ (by association)} \\
&= \left\lceil \frac{n+1}{2} \right\rceil \text{ (by substitution)}
\end{aligned}$$

Thus, since a number  $n \in \mathbb{N}$  can only be even or odd, we can conclude that for any  $n \in \mathbb{N}$ ,  $\left\lfloor \frac{n}{2} \right\rfloor + 1 = \left\lceil \frac{n+1}{2} \right\rceil$ .

**PROBLEM 1-c.** Let  $D(n) = T(n+1) - T(n)$ . Determine a recurrence for  $D(n)$ .

**Solution.**

Let  $D(n) = T(n+1) - T(n)$ . If we let  $n = 1$  be the base case for the recurrence as in  $T(n)$ , we

obtain the following:

$$\begin{aligned}
D(1) &= T(2) - T(1) \text{ (by definition of } D(n)) \\
&= T\left(\left\lfloor \frac{2}{2} \right\rfloor\right) + T\left(\left\lceil \frac{2}{2} \right\rceil\right) + 2 - 0 \text{ (by definition of } T(n) \text{ and } T(1)) \\
&= T(1) + T(1) + 2 \\
&= 2 \text{ (by definition of } T(1))
\end{aligned}$$

Thus, we can see that  $D(1) = 2$ . We now seek the general case for  $D(n)$  by expanding the its representation using the definition for  $T(n)$ , as shown below.

$$\begin{aligned}
D(n) &= T(n+1) - T(n) \text{ (by definition of } D(n)) \\
&= \left( T\left(\left\lfloor \frac{n+1}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n+1}{2} \right\rceil\right) + (n+1) \right) - \left( T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + n \right) \text{ (by definition of } T(n)) \\
&= T\left(\left\lceil \frac{n+1}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n+1}{2} \right\rfloor\right) - T\left(\left\lceil \frac{n}{2} \right\rceil\right) - T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 \\
&= T\left(\left\lceil \frac{n+1}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 \text{ (by the fact that } \lfloor (n+1)/2 \rfloor = \lfloor n/2 \rfloor) \\
&= T\left(\left\lfloor \frac{n}{2} \right\rfloor + 1\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 \text{ (by the fact that } \lceil (n+1)/2 \rceil = \lfloor n/2 \rfloor + 1)
\end{aligned}$$

Now, by observing that this last expression takes the same form as  $D(n)$ , we obtain the following:

$$\begin{aligned}
D(n) &= T\left(\left\lfloor \frac{n}{2} \right\rfloor + 1\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 \\
&= D\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1
\end{aligned}$$

Now, putting these results together, we obtain the following recurrence for  $D(n)$ :

$$\begin{aligned}
D(1) &= 2 \\
D(n) &= D\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1
\end{aligned}$$

**PROBLEM 1-d.** *Prove using the strong form of induction that for any  $n \in \mathbb{N}$ , if  $n \geq 1$  then  $D(n) = \lfloor \lg(n) \rfloor + 2$ .*

**Solution. Base ( $n = 1$ )**

By the definition of  $D(n)$ , we know the following:

$$\begin{aligned}
D(1) &= 2 \text{ (by definition of } D(n)) \\
&= 0 + 2 \text{ (by additive identity)} \\
&= \lg(1) + 2 \text{ (by property of logarithms)}
\end{aligned}$$

$$= \lfloor \lg(1) \rfloor + 2 \text{ (by property of floor function)}$$

Thus,  $D(1) = \lfloor \lg(1) \rfloor + 2$ , as desired.

**Induction** ( $n > 1$ )

Assume that  $D(k) = \lfloor \lg(k) \rfloor + 2$  for some  $k \in \mathbb{N}$  such that  $2 \leq k < n$ . We now show that  $D(n) = \lfloor \lg(n) \rfloor + 2$ .

$$\begin{aligned} D(n) &= D\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 \text{ (by definition of } D(n)) \\ &= \left(\left\lfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \right\rfloor + 2\right) + 1 \text{ (by induction hypothesis)} \\ &= \left(\left\lfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \right\rfloor + 1\right) + 2 \text{ (by commutative property)} \\ &= \left(\left\lfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \right\rfloor + \lg(2)\right) + 2 \text{ (by property of logarithms)} \\ &= \left(\left\lfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \lg(2) \right\rfloor\right) + 2 \text{ (by property of floor function)} \\ &= \left(\left\lfloor \lg(2 \left\lfloor \frac{n}{2} \right\rfloor) \right\rfloor\right) + 2 \text{ (by property of logarithms)} \\ &= \lfloor \lg(n) \rfloor + 2 \text{ (by the fact that } \lfloor \lg(2 \lfloor n/2 \rfloor) \rfloor = \lfloor \lg n \rfloor) \end{aligned}$$

Thus,  $D(n) = \lfloor \lg(n) \rfloor + 2$ , as desired.

**PROBLEM 1-e.** *Then prove that  $T(n) - T(1) = \sum_{k=1}^{n-1} D(k)$ , and show that an immediate consequence is that  $T(n) = \sum_{k=1}^{n-1} \lfloor \lg(k) \rfloor + 2$ .*

**Solution.** By the definition of  $D(n)$ , we observe the following:

$$\begin{aligned} \sum_{k=1}^{n-1} D(k) &= \sum_{k=1}^{n-1} (T(k+1) - T(k)) \\ &= (T(2) - T(1)) + (T(3) - T(2)) + \dots + (T(n-1) - T(n-2)) + (T(n) - T(n-1)) \\ &= T(n) - T(1) \end{aligned}$$

Therefore, since  $\sum_{k=1}^{n-1} D(k)$  turns into a telescoping sum, we see that it collapses to  $T(n) - T(1)$ , and since  $D(n) = \lfloor \lg(n) \rfloor + 2$ , we also know the following:

$$\begin{aligned} T(n) - T(1) &= \sum_{k=1}^{n-1} D(k) \\ &= \sum_{k=1}^{n-1} (\lfloor \lg(k) \rfloor + 2) \end{aligned}$$

Thus, we see that  $T(n) - T(1) = \sum_{k=1}^{n-1} (\lfloor \lg(k) \rfloor + 2)$ , and since  $T(1) = 0$ , we conclude that  $T(n) = \sum_{k=1}^{n-1} (\lfloor \lg(k) \rfloor + 2)$ .

**PROBLEM 1-f.** Now show that  $T(n) = \sum_{k=1}^{n-1} \lfloor \lg(k) \rfloor + 2$  implies that  $T(n) = O(n \log(n))$ .

**Solution.** Using the fact that  $T(n) = \sum_{k=1}^{n-1} (\lfloor \lg(k) \rfloor + 2)$ . We now evaluate this summation as follows:

$$\begin{aligned}
 T(n) &= \sum_{k=1}^{n-1} (\lfloor \lg(k) \rfloor + 2) \text{ (by definition of } T(n)) \\
 &= \sum_{k=1}^{n-1} \lfloor \lg(k) \rfloor + \sum_{k=1}^{n-1} 2 \text{ (by splitting the summation)} \\
 &< \sum_{k=1}^{n-1} \lg(n) + \sum_{k=1}^{n-1} 2 \text{ (by property of logarithms)} \\
 &= (n-1)\lg(n) + 2(n-1) \text{ (by evaluating each summation)} \\
 &= n\lg(n) - \lg(n) + 2n - 1 \text{ (by distribution)} \\
 &= O(n\lg(n)) \text{ (by definition of } O(f(n))) \\
 &= O(n \log(n)) \text{ (by definition of } O(f(n)))
 \end{aligned}$$

Thus, we can conclude that  $T(n) = O(n \log(n))$ .

**PROBLEM 2-a.**

**Solution.** This algorithm relies on the sorted nature of the sequence S to check for a pair of numbers by traversing from both endpoints of the sequence (front and back) towards the middle. At each iteration, the algorithm checks the pair of numbers at the front and back positions to see if they sum to the target value and, if not, advances the front pointer if the sum is less than the target or decrements the back pointer if the sum is more than the target. Since the algorithm only traverses through the list at most once before it terminates, the know the resulting time complexity must be  $O(n)$ .

The source code for this algorithm is shown below.

```

1 def sortedHasSum(S, x):
2     start = 0
3     end = len(S) - 1
4     while (start < end):
5         sum = S[start] + S[end]
6         if (sum < x):
7             start = start + 1
8         elif (x < sum):
9             end = end - 1

```

```

10         else:
11             return True
12     return False

```

### PROBLEM 2-b.

**Solution.** This function relies on merge sort (which always runs in  $O(n \log n)$  time) to put the sequence  $S$  in sorted order and then runs the same sortedHasSum routine again. Since the time complexity of sortedHasSum is  $O(n)$  and merge sort is  $O(n \log n)$ , we know the time complexity  $T(n)$  of hasSum( $S, x$ ), where  $n = |S|$ , is  $T(n) = O(n \log n) + O(n) = O(n \log n)$ .

The source code for this algorithm is shown below.

```

1 def hasSum(S, x):
2     S = mergeSort(S)
3     return sortedHasSum(S, x)

```

### PROBLEM 3.

**Solution.** Tail recursion was implemented by making the observation that only one 'recursive' call is needed after each partition of the target sequence. Therefore, by taking the larger of the two partitions and placing it on the stack, and then continuing with the smaller partition by adjusting the sequence bounds (either moving up the start index or moving down the end index), we make use of tail-recursion by only half of the sequence at a time. This tail-recursive implementation guarantees that the stack is always  $O(\log n)$ .

The source code for this implementation is shown below.

```

1 def qsort(A, p, r):
2     stack = list()
3     bottom = (0, -1)
4     stack.append(bottom)
5     while (len(stack) != 0):
6         while (p < r):
7             q = partition(A, p, r)
8             if (q - p + 1 < r - q):
9                 pair = (q + 1, r)
10                stack.append(pair)
11                r = q - 1
12            else:
13                pair = (p, q - 1)
14                stack.append(pair)
15                p = q + 1
16        newBound = stack.pop(len(stack) - 1)
17        p = newBound[0]
18        r = newBound[1]
19    return A

```

**PROBLEM Extra credit.** *Prove that for any  $n \in \mathbb{N}$ ,  $\lfloor \lg(2\lfloor n/2 \rfloor) \rfloor = \lfloor \lg(n) \rfloor$ .*

**Solution.**

**Case 1:**  $n$  is even ( $2 \mid n$ , or  $n = 2m$  for some  $m \in \mathbb{N}$ )

$$\begin{aligned}\lfloor \lg(2\lfloor n/2 \rfloor) \rfloor &= \lfloor \lg(2\lfloor 2m/2 \rfloor) \rfloor \\ &= \lfloor \lg(2\lfloor m \rfloor) \rfloor \\ &= \lfloor \lg(2m) \rfloor \\ &= \lfloor \lg(n) \rfloor\end{aligned}$$

**Case 2:**  $n$  is odd ( $2 \nmid n$ , or  $n = 2m + 1$  for some  $m \in \mathbb{N}$ )

$$\begin{aligned}\lfloor \lg(2\lfloor n/2 \rfloor) \rfloor &= \lfloor \lg(2\lfloor (2m + 1)/2 \rfloor) \rfloor \\ &= \lfloor \lg(2\lfloor (2m/2) + (1/2) \rfloor) \rfloor \\ &= \lfloor \lg(2\lfloor m + (1/2) \rfloor) \rfloor \\ &= \lfloor \lg(2(m + \lfloor 1/2 \rfloor)) \rfloor \\ &= \lfloor \lg(2m + 2\lfloor 1/2 \rfloor) \rfloor \\ &= \lfloor \lg(2m + 1) \rfloor \\ &= \lfloor \lg(n) \rfloor\end{aligned}$$