CS41: Algorithms In Class, September 6

Lab 1

Lecturer: Joshua Brody

Below is an example of a solid solution to the cowpath problem, written in latex. Text written ⟨*in this font*⟩ are instructions to you—they tell you what parts are important/optional/etc. Of course, you do not *have* to latex your solutions this way (or at all). However, if you're not sure what to do, this is a good structure to have.

1. **The Cowpath Problem.** You and a friend are to meet on the Appalachian Trail (AT). Unfortunately, you know very little about the AT. You are now on the AT, and you know your friend waits for you *somewhere* on the AT, but you do not know where. Design and analyze an algorithm to reach your friend. Suppose your friend is m miles away from you (but you don't know m, nor do you know in which direction your friend is) How far do you travel?

   Solution.

   ⟨*Put what you think is the main challenge of the problem here.*⟩

   We have no way of knowing where our friend is, except by running into him. The challenge is that we don't know whether our friend is north or south, and if he is currently close to us, we don't want to walk all the way in the wrong direction. On the other hand, if we keep walking back and forth along the trail, we cover the same ground over and over. When m is large, we end up walking a long way before meeting our friend.

   Idea:

   ⟨*In this section, you should write one or two sentences about your high-level approach. This part is optional but can help the grader understand what you're trying to do.*⟩

   Iteratively walk north and south along the trail, *doubling* the distance you walk each iteration. This way, you cover distance in each direction, but you don't travel distance you've covered before *too much.*

   Algorithm:

   ⟨*put pseudocode here.*⟩

```
Cowpath
arrived = FALSE
k = 1
while !arrived and !South-end and !North-end
      // break out of loop as soon as terminal condition met
      WALK(k,NORTH) // walk k miles NORTH
      WALK(2k,SOUTH)
      WALK(k, NORTH)
      k = 2k // double length for next iteration
if !arrived
      if SouthEnd
            WALK NORTH until arrived
      else WALK SOUTH until arrived
else // We have arrived and are happy.
```

⟨*The* WALK *procedure is thorough but probably overkill. Just stating that you break out of the while loop after reaching friend or hitting the end of the trail should be enough.*⟩

```
WALK(k, d)
if k == 0
      return !STOP
walk one mile in direction d
if you meet your friend
      arrived = TRUE
      return STOP
if you reach NORTH end of trail
      North-end = TRUE
      return STOP
if you reach SOUTH end of trail
      South-end = TRUE
      return STOP
return WALK(k-1,d)
```

Analysis. ⟨*In this part, you should (a) prove correctness of the algorithm and (b) analyze the efficiency. For the cowpath problem, the proof of correctness is basically trivial. For other problems, the proof-of-correctness is the hard part, and the efficiency analysis is easy*⟩

Claim 1. The algorithm COWPATH eventually halts when we meet our friend.

Proof. We clearly keep walking longer and longer distances, eventually covering the entire trail until we arrive at our friend. □

Theorem 1. *If our friend is m miles away, then we walk $O(m)$ miles during the execution of* COWPATH*.*

Proof. First, consider the case where we arrive before ever reaching the end of the trail. During the ith full iteration of the while loop, we travel $4k$ miles, where $k = 2^i$. We execute the while loop until $k \geq m$. In the final iteration, we travel (a) m miles if our friend waits to the north of us, and (b) $(2k + m)$ miles if our friend waits to the south. Let $t := \lceil \log m \rceil$. Then, the total distance traveled is at most

$$\begin{aligned} 2 \cdot 2^t + m + \sum_{i=1}^{t-1} 4(2^i) &\leq 2 \cdot 2^{1+\log m} + m + 4 \sum_{i=1}^{t-1} 2^i \\ &\leq 5m + 4(2^t - 1) \\ &\leq 5m + 4 \cdot 2m \\ &< 13m = O(m) \end{aligned}$$

Next, consider the case where we reach the southern terminus of the trail before arriving. Let ℓ be the distance between the southern end and the starting point, and let t be the smallest integer such that $2^t \geq \ell$. Fix $k := 2^t$. Note that $k < m$, since otherwise we would reach our friend before hitting the southern end of the trail. We also know that $\ell \leq k$ since otherwise we would continue to iterate.

We travel $4 \sum_{i=1}^{t-1} 2^i$ miles during full iterations of the while loop. In the last iteration (when we hit the southern end of the trail), we travel $2k+\ell$ miles. In addition, we travel an additional $\ell + m$ miles before arriving at our friend. In total, we travel

$$\begin{aligned} (2k + \ell) + (\ell + m) + 4 \sum_{i=1}^{t-1} 2^i &\leq 2k + 2\ell + m + 4 \cdot 2^t \\ &\leq 5m + 4k \\ &= 9m = O(m) \ . \end{aligned}$$

The case where we hit the northen end of the trail before arriving at our friend is similar. In all cases, we travel $O(m)$ miles. Thus, the proof is complete. $\square$