Problem Set 4

Frederick Meneses

April 2021

Question 1: Nested Loops

(a) Starting with the innermost loop:
Loop 3's body takes 1 step and iterates $\lceil \frac{n}{2} \rceil$ times.
Loop 2's body takes Loop 3's steps and iterates $\lfloor \log n \rfloor$ times.
Loop 1's body takes Loop 2's steps and iterates $\lceil \log n \rceil$ times.

Therefore, the total time is $1 + (\lceil \log n \rceil \cdot \lfloor \log n \rfloor \cdot \lceil \frac{n}{2} \rceil)$, which is $\Theta(n \log^2 n)$

(b) Splitting into cases:
Case 1: i is odd
Loop 2's body takes 1 step and iterates from j to 0, which is i steps.
Case 2: i is even
Loop 3's body takes 1 step and iterates from j to n, which is n - i steps.

Examining outermost loop: Loop 1's body takes the cost of (Loop 2 for all odd i) + (Loop 3 for all even i) + 2 (for the assignment statements). It iterates n times.

$$2 + \sum_{i=0}^{\frac{n-1}{2}} i + \sum_{i=1}^{\frac{n-1}{2}} (n - i) = 2 + \sum_{i=0}^{n-1} i = 2 + \frac{n(n-1)}{2}$$

Therefore, the total time is in $\Theta(n^2)$.

(c) From 1b we know that the cost of the body of Loop 1 is i, which is exactly the number of print statements.

Loop 1 iterations:

$$\sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$$

Question 2: Worst Case Analysis

(a) $\forall n \in \mathbb{N}$, let n = len(lst).
Loop 2 iterates at most i times.

Loop 1 iterates at most n times.
Therefore the running time is,

$$\sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$$

which is $\mathcal{O}(n^2)$.

(b) Given an input family: s = 2, lst = [0, 0, ..., 1, 1, 0]
$\forall n \in \mathbb{N}$, let n = len(lst)

Loop 1 iterates at most n-1 times.

$$\sum_{i=0}^{n-2} = \frac{(n-1)(n-2)}{2}$$

This has an early return, therefore the running time is $\Omega(n^2)$.

(c) To find an input family that is len(lst),

$$\sum_{i=0}^{x} \frac{x(x-1)}{2} = \text{len(lst)}, \forall x \in \mathbb{N}$$

Let $n \in \mathbb{N}$ and n = len(lst).
Let $x = \sqrt{n}$

The function must terminate when the length of the list is a positive square number x^2 and so the running time is $\Theta(n)$.

Question 3: Worst and Best Case Analysis

(a) Upper Bound:
Loop 3's body takes 1 step and iterates j - i times at most.
Loop 2's body takes Loop 3's steps and iterates n - 1 - i times at most.
Loop 4's body takes 1 step and iterates n - 1 - i times at most.
if not lst[j] ≤ 0, Loop 5's body takes 1 step and iterates n - i - 1 times at most.
$\forall n \in \mathbb{N}$ n = len(lst),
Total time is

$$(n-1)(2(n-1-i) + \sum_{j=i+1}^{n} (j-i))$$

which is $\mathcal{O}(n^3)$
Lower Bound: lst = [0, 2, 4, ..., 2n]. Thus, the worst case is $\Theta(n^3)$.

(b) Lower Bound:
$\forall n \in \mathbb{N}$, n = len(lst). The function cost = 2, which is $\Omega(1)$.
Upper Bound:
lst = [-1, -1, ..., -1]. The function cost is constant, which is $\mathcal{O}(1)$. Thus, the best case is $\Theta(1)$.