

5. Ternary Search is a modification to Binary Search that splits the search region into thirds rather than halves. Here is the pseudocode for Ternary Search:

[15 pts]

```
\\ PRE: A is a sorted list of length n.
\\ PRE: TARGET is a target element.
function ternarysearch(A, TARGET)
    L = 0
    R = n-1
    while (L <= R)
        C1 = floor((2L+R)/3)
        C2 = floor((L+2R)/3)
        if A[C1] == TARGET
            return C1
        elif A[C2] == TARGET
            return C2
        elif TARGET < A[C1]
            R = C1 - 1
        elif TARGET > A[C1]
            if TARGET < A[C2]
                L = C1 + 1
                R = C2 - 1
            elif TARGET > A[C2]
                L = C2 + 1
            end
        end
    end
    return FAIL
end
\\ POST: Value returned is either the index or FAIL.
```

Assume initializing L and R combined takes time c_1 , the body of the while loop takes time c_2 (even if it returns midway through), and returning FAIL takes time c_3 . Assume also that $n = 3^N - 1$ for some N , which ensures clean divisions. Finally, you are given that

$$\sum_{i=0}^{n-1} 3^i = \frac{1}{2}(3^n - 1)$$

$$\sum_{i=1}^n i3^i = \frac{1}{4}(2n \cdot 3^n - 3^n + 1)$$

Compute the average-case runtime of Ternary Search and state its time complexity. Then, state whether the time complexity is better, worse, or equal to that of Binary Search's average case.