CS381 Homework 1 Problem 2

Connor Couetil

January 29, 2026

1 Exercise 3.7

The intuition is to compute pairs of sums, and then to realize that recursively computing pairs of sums gives you the even prefix sums, after which the odd prefix sums can be computed by simple subtraction of the previous value

`prefix-sum(`A`)`: Given a array A of length n, it returns an array of length n containing the prefix sums of A.

```
If $|A|$ is 1, return $A$
Let $B$ be an array of length $\lfloor n/2 \rfloor$
$B[i] = a_{2i} + a_{2i-1}$ for $i = 1, ..., \lfloor n/2 \rfloor$
$C =$ prefix-sum($B$)
Let $D$ be a new array of length $n$
for $i = 1, ..., n$
    if $i$ is even, $D[i] = C[i/2]$
    if $i$ is odd, $D[i] = C[\lceil i/2 \rceil] - A[i]$
return $D$
```

The recurrence relation for work will be notated with W, the recurrence relation for time will be notated with T.

The total work performed is $O(n\,log(n))$

$$W(1) = 1$$
$$W(n) = O(n) + W(n/2) + O(n)$$

The total time taken is $O(log(n))$

$$T(1) = 1$$
$$T(n) = O(1) + T(n/2) + O(1)$$

The summations can be done in parallel, thus $O(n)$ work is done in $O(1)$ time. The size of the array is cut down by half at each iteration, thus the height of the tree is $log(n)$ with constant time taken at each level.