

2.1-4

Input: Two arrays of size n , representing two n -digit binary integers. Each array cell contains either a 1 or 0 and represents $\text{array}[i] \times 2^i$ in decimal value.

Output: One array of length $n+1$ whose cells contain the result of performing binary addition element by element on the input arrays.

function Add(A, B)

 carry = 0, output = Array(length = A.length + 1)

 for $i = 0$ until $i = A.length$

 sum = A[i] + B[i] + carry

 if sum ≥ 1

 carry = 1

 sum = sum % 2

 else

 carry = 0

 output[i] = sum

 output[A.length] = carry

 return output

2.2-3

Average Case: $n/2$ because half of the time the target element will be found before the median and half after

Worst Case: n . If the target is the last element or not in the list at all, the algorithm checks n elements

Average Case & Worst Case: $O(n)$. n is the most significant variable in both cases runtime equations and constants do not matter