a1

Tony and Andrew

January 2024

1. **Solution:**

1.

a) Worst case $\mathcal{O}(n^2)$ time

Proof: We will prove that for every input of size $n \geq 2$, the number of steps executed by the algorithm is no larger than cn^2, where $c \in \mathbb{R}^{>0}$ is some constant.

Lines 1 and 2 involve assignment operations, as such, they each take constant time. We can say that the total of all such constant time operations is $\theta(1)$.

For any input array, line 3 "loops" exactly $n-1$ times.

Since $i \leq n-1$ (by the for loop on line 3), the for loop on line 4-5 iterates no greater than $n-2$ times, with each iteration taking constant time. Therefore, no greater than $n-2$ steps for each pass from lines 4-5 will be taken.

Line 6 assignment takes one step, so lines 4 to 7 take no more than $(n-1)$ steps after one iteration.

So the for loop described in line 3 iterates $n-1$ times and lines 4 to 7 are nested in the for loop on line 3. Thus, total steps $\leq (n-1)(n-1)+1$ where the 1 accounts for the constant time of steps 1-2.

With the maximum total number of steps for this algorithm, we want to show an upper bound on the runtime.

Let $c = 1$, and let $n_0 = 2$.

$$\begin{aligned}
\text{Assuming } n \geq n_0: \quad & (n-1)(n-1)+1 \\
& = n^2 - 2n \\
& \leq n^2 \\
& = cn^2
\end{aligned}$$

So number of steps is no greater than $(n-1)(n-1)+1 \in \mathcal{O}(n^2)$. In other words the worst case is in big-Oh of n^2.

b) Worst case $\Omega(n^2)$ time

Proof: we posit that there exists an input of size $n \in \mathbb{N}$ where $n \geq 2$, such that the number of steps executed is no smaller than cn^2, where $c > 0$ and c is a real number.

Consider an input A of size n with n "1"s, i.e., $[1, 1, \ldots, 1]$. By definition, each element in the array is 1. In other words, for all $k \in \mathbb{N}$ such that $0 \leq k \leq n-1$, $A[k] = 1$

After every iteration of the loop starting on line 3, elements of indices 0 to $i-2$ have been modified, while the rest of the elements (including A[i] and A[i-1]) are untouched and equal to one. Proof of this by induction on "i", the for loop variable specified on line 3 that can take on integer values between 0 to n-1 inclusive:

- Base case: Consider the first iteration of the loop starting on line 3, where i = 1. The loop starting on line 4 does not execute, so by the end of line 6, no contents of the array have been altered, and all elements are equal to 1.
- Inductive step: Let i be an arbitrary integer between 2 to n, and assume that indices 0 to i-3 have been altered, while the rest are equal to 1 (the inductive hypothesis). After the i^{th} iteration of the loop starting on line 3, the loop on lines 4-5 will have just modified the values of indices 0 to i-2. Therefore, the only indices that have values not equivalent to 1 will be indices i-2 and lower, while all the other indices have values equal to 1.

Therefore, for the comparison at line 6, $A[i]$ is always equal to $A[i-1]$ (encodes "1"), and no early return happens.

For each iteration i of the outer loop, the inner loop on lines 4-5 iterates i-1 times. Since line 6 takes constant time, we can say the the total number of steps of the body from lines 4-7 is i-1+1 = i.

Lets note the number of steps within each iteration of the loop on line 3 for i=1,2, up to n-1.

When $i = 1$, one step occurs from line 4 to 7 (one comparison on line 6).

When $i = 2$, two steps occur from line 4 to 7 (one execution of line 5, one on 6).

When $i = n-1$, $n-1$ steps occur (n-2 executions of line 5, one on line 6).

Adding the steps from line 4 to 7 for each iteration of the loop on line 3, it follows that the total number of steps is $1 + (1 + 2 + \ldots + n - 1)$.

$$1 + (1 + 2 + \ldots + n - 1) = 1 + \frac{n(n-1)}{2}$$

Let $c = \frac{1}{4}$. Since $n \geq 2$, $n^2 \geq 2n$.

$$\begin{aligned}
&\text{So } n^2 - 2n \geq 0 \\
&\text{So } n^2 - 2n + 4 \geq 0 \\
&\text{So } \frac{1}{4}n^2 - \frac{1}{2}n + 1 \geq 0 \\
&\text{So } \frac{1}{2}n^2 - \frac{1}{2}n + 1 \geq \frac{1}{4}n^2
\end{aligned}$$

$$\boxed{\frac{n(n-1)}{2} + 1 \leq cn^2}$$

So, total number of steps in the worst case $\in \Omega(n^2)$.

2. **Solution:**

2.

Let the random variable R be the number of array accesses given an non-empty input array A where the specifications (A contains only 0's and 1's, the 1's appear before all the zeros, and the size of A is $n = 3m$ for some $m \in \mathbb{N}$) are ensured.

We can define indicator random variables W_i for $i \in \{0, 1, \ldots, n-1\}$ where $W_i = 1$ if an array access on line 4 is executed for index i, and $W_i = 0$ otherwise. From inspection, only indices equal to $3k - 1$ for some $k \in \mathbb{N}$ can be accessed by line 4, since the loop starts on index 2 and increments by 3 upwards. We'll name that set of indices S.

Let x be an arbitrary index from S. Consider two cases:

- $x = 2$. Then $P(W_x = 1) = 1$, since for any input, index 2 is always accessed first via line 4.
- $x \neq 2$. The index x is accessed on line 4 if and only if $A[x - 3] = 1$. The number of unique possibilities where this occurs is $n - 1 - (x - 3) + 1 = n - x + 3$, equivalent to counting the cases where the last 1 is in index $x - 3$, $x - 2$, $\ldots$, all the way to $n - 1$. Also, the sample space has size $n + 1$ (considering the case where all are 0's, all the way to 1's). Since each possibility is equally likely, $P(W_x = 1) = \frac{n-x+3}{n+1}$.

$P(W_x = 1)$ for case 1 actually follows the same formula as on case 2 since $\frac{n-(2)+3}{n+1} = 1$.

From the above, we can see that:

$$\begin{aligned} E\left(\sum_{i=0}^{n-1} W_i\right) &= E\left(\sum_{k=1}^{n/3} W_{3k-1}\right) \\ &= \sum_{k=1}^{n/3} P(W_{3k-1} = 1) \\ &= \sum_{k=1}^{n/3} \left(\frac{n-(3k-1)+3}{n+1}\right) \end{aligned}$$

This outcome will be useful for later when we determine $E[R]$.

We can also define indicator random variables X_i for $i \in \{0, \dots, n-1\}$, where $X_i = 1$ if an array access on line 7 is called for index i, and $X_i = 0$ otherwise. To find $P(X_i = 1)$, we consider three cases:

- $i = 3k$, where $k \in \mathbb{N}$. There are 3 possibilities where index i is accessed via line 7; if the first 0 is in index i, if it's in index $i+1$ or index $i+2$. Therefore,

  $$P(X_i = 1) = \frac{3}{n+1},$$

  with the denominator representing the size of the sample space.
- $i = 3k-2$, where $k \in \mathbb{N}$. There are 2 possibilities where index i is accessed via line 7; if the first 0 is in index i, or $i+1$. Therefore,

  $$P(X_i = 1) = \frac{2}{n+1},$$
- $i = 3k-1$, where $k \in \mathbb{N}$. There is one possibility where index i is accessed via line 7; if the first 0 is in index i. Therefore,

  $$P(X_i = 1) = \frac{1}{n+1}.$$

We note that $R = \sum_{i=0}^{n-1} W_i + \sum_{j=0}^{n-1} X_j$. This is contingent on the observation that each index can be accessed at most once by line 4, and at most once by line 7. The value of W_i, defined as 1 if the index i is accessed via line 4, is therefore equivalent to the number of times index i is accessed via line 4. This is similar for X_j, so the summation of these random variables is equal to the total number of array accesses. Therefore, to find $E[R]$, we see:

$$
\begin{aligned}
E(R) &= E\left(\sum_{i=0}^{n-1} W_i + \sum_{j=0}^{n-1} X_j\right) \\
&= E\left(\sum_{i=0}^{n-1} W_i\right) + E\left(\sum_{j=0}^{n-1} X_j\right)
\end{aligned}
$$

From before, we know

$$
E\left(\sum_{i=0}^{n-1} W_i\right) = \sum_{k=1}^{n/3}\left(\frac{n-(3k-1)+3}{n+1}\right).
$$

In contrast,

$$
\begin{aligned}
E\left(\sum_{j=0}^{n-1} X_j\right) &= \sum_{j=0}^{n-1} P(X_j = 1) \\
&= \sum_{k_1=0}^{n/3} P(X_{3k_1} = 1) + \sum_{k_2=1}^{n/3} P(X_{3k_2-2} = 1) + \sum_{k_3=1}^{n/3} P(X_{3k_3-1} = 1) \\
&= \left(\frac{n}{3}\right)\left(\frac{3}{n+1}\right) + \left(\frac{n}{3}\right)\left(\frac{2}{n+1}\right) + \left(\frac{n}{3}\right)\left(\frac{1}{n+1}\right)
\end{aligned}
$$

So,

$$
\begin{aligned}
E(R) = &\sum_{k=1}^{n/3}\left(\frac{n-(3k-1)+3}{n+1}\right) \\
&+ \left(\frac{n}{3}\right)\left(\frac{3}{n+1}\right) + \left(\frac{n}{3}\right)\left(\frac{2}{n+1}\right) + \left(\frac{n}{3}\right)\left(\frac{1}{n+1}\right)
\end{aligned}
$$