

# CSC165H1 Problem Set 4 (due 12/6/17)

Jacob Nazarenko, James Currier, Mark Abdullah

December 20, 2017

1. a) Want to show that for any graph  $G = (V, E)$ , the number of vertices with odd degree is even.

Let  $P(n) : \forall G = (V, E), |V| = n \Rightarrow |\{v \in V \mid d(v) \text{ is odd}\}|$  is even

**Claim:**  $\forall n \in \mathbb{N}, P(n)$

**Proof (by induction):**

Base cases: For  $n = 0$ , the graph has no vertices, and therefore there are an even number (which is 0) of vertices with an odd degree. For  $n = 1$ , there can be no edges in the graph, and therefore the number of vertices with odd degree (which is still 0) is still even.

Inductive step: Let  $k$  be an arbitrary natural number, and assume that  $P(k + 1)$  holds, that is, assume that for an arbitrary graph  $G = (V, E)$  with  $k + 1$  vertices, the number of vertices with odd degree is even. We want to show that  $P(k)$  holds, that is, that the number of vertices with odd degree in an arbitrary graph with  $k$  vertices is even. Now, let us take a graph with  $k + 1$  vertices, and remove one of the vertices from this graph. This vertex may have had either an even or an odd degree. The following two sets of cases follow:

In the first set of cases, we assume that the vertex removed had an even degree. This implies that either (a) it was connected to an even number of vertices with odd degree and an even number of vertices with even degree, or (b) it was connected to an odd number of vertices with odd degree and an odd number of vertices with even degree.

Case 1a: In this case, the vertex removed was connected to an even number of vertices with odd degree and an even number of vertices with even degree. One must notice that when the vertex is removed (along with all of the edges connected to it), all of the vertices connected to it with even degree switch to an odd degree and all vertices connected to it with an odd degree switch to an even degree (in all cases), as each of the vertices has an edge removed from it. By this observation, we may conclude that if this vertex is removed, then there will still be even numbers of vertices with even and odd degrees among the vertices that were previously connected to it (because the even and odd sets of vertices have essentially 'flipped'). Therefore, combining this with our inductive hypothesis (which stated that there was an even number of vertices with odd degree to start), there will still be an even number of vertices with odd degree in total.

Case 1b: In this case, the vertex removed was connected to an odd number of vertices with odd degree and an odd number of vertices with even degree. This means that the number of vertices with odd degree that are not connected to the removed vertex is also odd (because odd + odd makes even according to Example 2.18 from the course notes). Based on our previous observation regarding the sets of vertices 'flipping,' we can say here that once again, the sets of vertices with odd and even de-

degrees connected to the removed vertex will flip, and you will end up with the same situation as before the removal - an odd number of vertices with odd degree and an odd number of vertices with even degree. Because the vertex being removed has an even degree, the number of vertices with odd degree in the whole graph remains an even number (because as previously stated, odd + odd makes even).

In the second cases, we assume that the vertex removed had an odd degree. This implies that when this vertex is removed, the graph loses an extra vertex with an odd degree, and also that the vertex is connected to either (a) an odd number of vertices with odd degree and an even number of vertices with even degree, or (b) an even number of vertices with odd degree and an odd number of vertices with even degree.

Case 2a: In this case we assume that the removed vertex was connected to an odd number of vertices with odd degree and an even number of vertices with even degree. Therefore there is an odd number of vertices with odd degree that are not connected to this vertex in the graph. When it is removed, and the sets of vertices connected to it 'flip,' you will have an even number of vertices with odd degree that were previously connected to it, in addition to the odd number (still the same) not originally connected to it, that remain in the graph. However, the vertex itself is subtracted from the total number, and this makes the number of vertices with odd degree remain even (because even + odd - 1 makes even).

Case 2b: In this final case we assume that the vertex was connected to an even number of vertices with odd degree and an odd number of vertices with even degree. This implies that there is an even number of vertices with odd degree that are not connected to the vertex removed in the graph. It also implies that when the vertex is removed and the sets of vertices connected to it are 'flipped,' we end up with an odd number of vertices with odd degree that used to be connected to the removed vertex, in addition to the even number of vertices with odd degree that were not originally connected to the removed vertex. Once again, taking into account the fact that the removed vertex itself had an odd degree, we once again end up with an even number of vertices with odd degree (because odd + even - 1 makes even).

We have now proven that regardless of the vertex removed, the number of vertices in the graph with odd degree will always remain even, and have thereby proven the original statement. ■

b) Let  $G = (V, E)$  with,  
 $V = \{v_1, v_2, v_3, v_4\}$   
 $E = \{(v_3, v_1), (v_3, v_2), (v_3, v_4), (v_2, v_4)\}$   
Then  $d(v_1) = 1 \wedge d(v_2) = 2 \wedge d(v_3) = 3$  as required.

Want to show this is the only possible graph that satisfies the properties as outlined in the question.

**Proof:** Since there are four vertices, and the degree of  $v_3$  is 3, it must be connected to every other vertex. Then the condition that vertex  $v_1$  has degree 1 will be satisfied as it is connected to  $v_3$ . The last condition to satisfy is that  $v_2$  has a degree of 2. To add an edge to  $v_2$ , it must be connected to  $v_4$  as  $v_1$  and  $v_3$  are already at the required degree. Then the only possible set of edges to satisfy these conditions is as described, which is equal to the set  $E$  defined above. ■

c) We want to show that

$$\forall G = (V, E), \forall n \in \mathbb{N}, (\exists v \in V, d(v) = n) \implies |V| \geq n + 1$$

**Proof:** Let  $G = (V, E)$

Let  $n \in \mathbb{N}$   
 Assume  $\exists v \in V, d(v) = n$   
 WTS  $|V| \geq n + 1$

Since the degree of  $v$  is equal to  $n$ , there must exist  $n$  other unique vertices  $\{v_0, v_1, \dots, v_{n-1}\}$  that are connected to  $v$ . Then the graph  $G$  has at least  $n + 1$  vertices. We've shown  $|V| \geq n + 1$  as needed. ■

d) We want to show that

$$\forall G = (V, E), (\forall v \in V, d(v) = 2) \Rightarrow (G \text{ has at least 1 cycle})$$

**Proof:** Let  $G$  be an arbitrary graph, and assume that the degree of every vertex in this graph is 2. We know based on this assumption that each vertex in the graph has exactly 2 neighbors. Now take an arbitrary vertex  $a \in V$  in this graph, and consider one of its two neighbors, vertex  $b$ . Let the edge  $e \in E$  be the edge between these two vertices. Now, consider the other edge connected to vertex  $a$ . If we follow the path of unique edges away from vertex  $a$  (that does not include edge  $e$ ), then you must eventually reach vertex  $b$ . This is true due to the following facts:

- First, each vertex in the path leading away from vertex  $a$  must have degree 2, and therefore there is only 1 path to take onto each next vertex in the sequence
- Second, none of the vertices in this sequence can connect back to any other vertex but vertex  $b$  itself, because we have already assumed that for each vertex other than  $b$ , both of the vertex's edges are already used up, and therefore connecting back to any vertex other than vertex  $b$  would make it so that one of the vertices in  $G$  would have a degree greater than 2.
- Lastly, the sequence of vertices leading away from vertex  $a$  cannot stop or break at any point, as this would imply that there is a vertex in  $G$  with a degree less than 2

We have thereby shown that there are 2 ways of going between neighbors  $a$  and  $b$ , and consequently that the graph  $G$  has at least 1 cycle (disconnected from any other cycles that may or may not exist). We have now proven the statement. ■

e) We want to show that

$$\forall G = (V, E), (|V| > 4 \wedge \forall v \in V, d(v) \geq |V| - 3) \Rightarrow (G \text{ is connected})$$

**Proof:** Let  $G = (V, E)$  be an arbitrary graph and assume  $n = |V| > 4$  and  $\forall v \in V, d(v) \geq |V| - 3$ . Assume that  $G$  is disconnected, and consequently that there are two groups of vertices in  $G$  that are not connected by any edges.<sup>1</sup> Let these two groups of vertices be represented by  $\beta$  and  $\beta'$ , and let  $|\beta| = a$  and  $|\beta'| = n - a$ . In other words, let there be  $a$  vertices in  $\beta$  and  $(n - a)$  vertices in  $\beta'$ . We know from our earlier assumptions that  $a < n$  (as there must be at least 1 vertex that is disconnected from the rest) and that  $n > 4$ , which in this context means that  $n \geq 5$ . However, in the case that  $n=5$ , for example, the degree of each vertex must be at least 2. If  $a = 1$ , this does not work, as the 1 vertex in  $\beta$  can have at most degree 0. If  $a = 2$ , then this also doesn't work, because the 2 vertices in  $\beta$  can have at most degree 1, which is less than the minimum of  $|V| - 3 = 2$  needed. When  $a = 3$ , this seems to work (as each vertex in  $\beta$  can have a maximum degree of 2), but in actuality it still doesn't, as we must also consider that the disconnected group  $\beta'$ , which has  $n - a = 5 - 3 = 2$  vertices won't work by our previous statement. Because larger values of  $|V|$  will require larger minimum degrees, we may therefore safely assume for graphs with  $n = |V| > 5$  vertices that  $a < n - 2$ . Any disconnected grouping of 3 vertices (each of which has a maximum degree of 2) will not work for graphs where

<sup>1</sup>We will assume for now that there are only two disconnected groups in this graph, and will show later on in the proof why the contradiction works with more than two disconnected groups.

$|V| > 5$ , because the maximum degree required will be greater than 2 (because  $|V| - 3 > 2$ ).

First, let us derive the degree-sum of the vertices in a set when they are connected by the maximal number of edges. We know that for any set of vertices  $V$ , the degree sum of the vertices is equal to

$$\sum_{v \in V} d(v) = 2|E| \quad (\text{where } |E| \text{ is the number of edges between the vertices})$$

We also know that the maximum number of edges between a set of vertices in a set of vertices where  $|V| = n$  is

$$\frac{n(n-1)}{2}$$

Combining these two facts, we have that the degree sum of set of vertices connected by the maximum number of edges is

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

Next, let us proceed by rearranging our assumption that  $a < n - 2$ :

$$\begin{aligned} a &< n - 2 \\ a + 2 &< n \\ a^2 + 2a &< an \\ a^2 - a &< an - 3a \\ a(a - 1) &< a(n - 3) \end{aligned}$$

The resulting inequality states that the degree-sum of the vertices in  $\beta$  when they are connect with the *maximum* number of edges possible is strictly less than the required degree-sum of the vertices given that  $\forall b \in \beta, d(b) = |V| - 3 = n - 3$ . This means that there can be no way that the degree of each vertex in the disjoint group of vertices  $\beta$ , even with the maximum number of edges between them, can be greater than or equal to  $|V| - 3$ , and therefore there must be edges that connect vertices from this group to vertices in the other group,  $\beta'$ . It can also be shown that the same situation applies to the group  $\beta'$ . Again, we know that the degree sum of the vertices in  $\beta'$  with the maximal number of edges between them is  $(n - a)(n - a - 1)$  by our previous definitions. Then we can rearrange the statement that  $a > 2$  to prove our point, as we know that if  $a$  must be greater than 3, then it must be greater than 2:

$$\begin{aligned} a &> 2 \\ a(n - a) &> 2(n - a) \\ an - a^2 &> 2n - 2a \\ a^2 &< an - 2n + 2a \\ n^2 + a^2 &< n^2 + an - 2n + 2a \\ n^2 - 2an + a^2 &< n^2 - an - 2n + 2a \\ n^2 - 2an + a^2 - n + a &< n^2 - an - 3n + 3a \\ (n - a)(n - a - 1) &< (n - a)(n - 3) \end{aligned}$$

The resulting inequality says that the degree sum of the maximally connected vertices in  $\beta'$  is less than the required degree sum, and therefore must be at least one more edge connecting  $\beta'$  to  $\beta$ . Now that we have proven this result with both groups, we may also draw the conclusion that if there are

more than 2 disconnected groups in the graph  $G$ , then it is impossible for the required degree-sum to be attained. This is because we have just proven that even 2 disjoint groups will not attain this required degree-sum, and breaking up the graph  $G$  into smaller disjoint groups of vertices will only make the maximum number of edges in each group smaller. We may now say that we have reached a contradiction, and that the graph  $G$  therefore must be connected. ■

f) Based on the result in the previous part, we can say that the degree-sum of all the vertices in the graph  $G$  must be

$$\text{degree-sum} \geq n(n-3) > 4$$

because  $n = |V| > 4$  and  $\forall v \in V, d(v) \geq |V| - 3$ . We know that the degree-sum counts all edges exactly twice, so the total number of edges should be

$$|E| \geq \frac{n(n-3)}{2} > 2$$

The results from examples 6.6 and 6.7 in the course notes state that there exists a graph with  $\frac{(n-1)(n-2)}{2}$  vertices that is not connected and that graphs with  $\frac{(n-1)(n-2)}{2} + 1$  vertices must be connected, but these examples do not restrict the number of vertices to be greater than 4, and do not restrict the degree of each vertex to be greater than or equal to  $n-3$ . That having been said, these two values are 1 and 2 more, respectively, than the lower bound on the number of edges obtained in the previous part. This is most likely because of the restrictions in this problem, but there does not seem to be any obvious connection between these results. We know for sure that the result of 6.7 should work in our case (because this value is clearly larger than the lower bound), but the existence quantifier in 6.6 makes it so that, without the restriction in the previous part, a graph that has that many vertices may be disconnected, despite the value being higher than the required number of edges found in the previous part.

## 2. a) **Proof (by contradiction)<sup>2</sup>:**

Suppose there are two different representations of a number in the form  $n = \sum_{i=0}^p b_i 2^i$ , and assume they are equal.

That is, suppose  $n = \sum_{i=0}^p b_i 2^i = \sum_{j=0}^s f_j 2^j$  and  $p \neq s \vee b \neq f$

Let us first prove the following lemma by induction on  $n$ :

$$P(n) : 2^n > \sum_{i=0}^{n-1} 2^i \quad \text{Claim: } \forall n \in \mathbb{N}^+, P(n)$$

Base case:  $2^1 = 2, \sum_{i=0}^0 2^i = 2^0 = 1, 2 > 1$

Inductive step: Now, let  $k$  be an arbitrary natural number, and assume  $P(k)$ , that is, that  $2^k > \sum_{i=0}^{k-1} 2^i$ . We now want to show that  $P(k+1)$  holds. We proceed as follows:

---

<sup>2</sup>For this proof, we assume that any binary number with leading zeroes should be considered without those leading zeroes. That is, the leading bit  $b_p$  in any number should be 1.

$$\begin{aligned}
2^{k+1} &= 2^k + 2^k \\
&> 2^k + \sum_{i=0}^{k-1} 2^i \\
&= \sum_{i=0}^k 2^i
\end{aligned}$$

We have thereby proven this lemma. ■

We're assuming in this problem that  $p = s$ :

Since  $p = s$ ,  $b \neq f$

So there must be at least one index<sup>3</sup>  $q$  where  $b_q \neq f_q$

Let  $q$  be the largest index where  $b_q \neq f_q$

So the part of the sum of  $b$  and  $f$  before this  $q$  is equal to  $\sum_{i=0}^{q-1} b_i 2^i$

Consider the remaining digits in  $b$  and the remaining digits in  $f$ . That is the part between  $q$  and the ends of  $b$  and  $f$ , inclusive.

Since we have assumed that both representations are equal to  $n$ , both of these parts should be equal in value.

However, since  $b_q$  and  $f_q$  differ, either  $b_q$  or  $f_q$  is 0, and the numbers represented by the remaining digits in  $b_q$  and  $f_q$ , if they are considered as standalone binary numbers, are not equal in value, because the greatest power of 2 in the number with the leading 1 is higher, and the value of this number is therefore higher by the result of the lemma proven above.

This leads us to a contradiction. ■

b) By Theorem 4.1, we know that every natural number has a binary representation.

By part (a) of this question, we know that every natural number greater than 0 has a unique binary representation.

We also know that the only way that 0 can be represented in binary is 0, as  $\forall n \in \mathbb{N}, 2^n > 0$ .

Therefore, there is a unique binary representation for every natural number.

The reason why it wasn't possible to make the domain of the previous proof "every number  $n \in \mathbb{N}$ ", is that -1 is the smallest integer such that  $2^{p+1} > n$ , and -1 is not a non-negative integer.

3. a) We want to show the average case running time of Search on the set of inputs as defined in Exercise 5.4 is  $\theta(1)$ .

**Proof:** As defined in exercise 5.4, define the set of inputs to be  $I_n$ , where for each input  $(lst, x) \in I_n$ ,  $lst$  has length  $n$ , and the search term  $x$ , and the elements of  $lst$  are all between the numbers 1 and 10.

Then the number of possible lists  $lst$  is equal to  $10^n$ , and the number of possible inputs of search is equal to  $10 \cdot 10^n = |I_n|$ , because we have 10 different search values of  $x$ .

---

<sup>3</sup>where indexing begins at the highest bit,  $b_p$ , and goes down decreasing  $p$  by 1

$$Avg_{Search}(n) = \frac{1}{|I_n|} \cdot \sum_{(lst, x) \in I_n} runtime\ of\ search(lst, x)$$

Define  $S_n$  as the set of all possible lists  $lst$ . We will instead fix a value of  $x$ , and find the running time for that fixed value.

$$Avg_{Search}(n) = \frac{1}{|I_n|} \cdot \sum_{x=1}^{10} \sum_{lst \in S_n} runtime\ of\ search(lst, x)$$

To calculate the total number of steps for a fixed  $x$ , we will split  $S_n$  based on the position that the first value of  $x$  appears, since the number of steps is the equal to the number of iterations the loops takes before finding  $x$ , or  $n$  iterations when it does not return early (if it does not find  $x$ ).

$$Avg_{Search}(n) = \frac{1}{|I_n|} \cdot \sum_{x=1}^{10} \sum_{i=0}^{n-1} (i+1) \cdot (\text{number of lists that have first } x \text{ in position } i)$$

We know the number of lists is  $10^n$  because there is 10 possible values for each index in the list. Then the number of lists with the first instance of a specific value in position 1 is  $1 \cdot 10^{n-1}$ . However this formula does not apply to every position since we are looking for lists that contain the first instance of a value in that position. For example, at position two, the possible number of lists is  $9 \cdot 1 \cdot 10^{n-2}$  since value in the first position can be any of 1-10, except for the value we are searching for, thus leaving 9 possibilities. Then the formula for the number of lists is as follows,

$$\begin{aligned} Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} \sum_{i=0}^{n-1} (i+1) \cdot 9^i \cdot 10^{n-1-i} \right) + n9^n \\ Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^{n-1} \cdot \sum_{i=0}^{n-1} (i+1) \cdot 9^i \cdot 10^{-i} \right) + n9^n \\ Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^{n-1} \cdot \sum_{i=0}^{n-1} (i+1) \cdot \left(\frac{9}{10}\right)^i \right) + n9^n \\ Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^{n-1} \cdot \left( \sum_{i=0}^{n-1} i \cdot \left(\frac{9}{10}\right)^i + \sum_{i=0}^{n-1} \left(\frac{9}{10}\right)^i \right) \right) + n9^n \end{aligned}$$

Using the given equation for a sum, and the equation for the sum of a geometric series, we can express the inner sums as follows,

$$\begin{aligned} Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^{n-1} \cdot \left( \frac{n \left(\frac{9}{10}\right)^n}{\frac{9}{10} - 1} + \frac{\frac{9}{10} - \left(\frac{9}{10}\right)^{n+1}}{\left(\frac{9}{10} - 1\right)^2} + \frac{1 - \left(\frac{9}{10}\right)^n}{1 - \frac{9}{10}} \right) \right) + n9^n \\ Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^{n-1} \cdot \left( \frac{n \left(\frac{9}{10}\right)^n}{\frac{-1}{10}} + \frac{\frac{9}{10} - \left(\frac{9}{10}\right)^{n+1}}{\frac{1}{100}} + \frac{1 - \left(\frac{9}{10}\right)^n}{\frac{1}{10}} \right) \right) + n9^n \end{aligned}$$

$$\begin{aligned}
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^{n-1} \cdot \left( -10n \left( \frac{9}{10} \right)^n + 100 \left( \frac{9}{10} - \left( \frac{9}{10} \right)^{n+1} \right) + 10 \left( 1 - \left( \frac{9}{10} \right)^n \right) \right) \right) + n9^n \\
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^{n-1} \cdot 10 \cdot \left( -n \left( \frac{9}{10} \right)^n + 10 \left( \frac{9}{10} - \left( \frac{9}{10} \right)^{n+1} \right) + 1 - \left( \frac{9}{10} \right)^n \right) \right) + n9^n \\
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^n \cdot \left( -n \left( \frac{9}{10} \right)^n + 9 - 10 \left( \frac{9}{10} \right)^{n+1} + 1 - \left( \frac{9}{10} \right)^n \right) \right) + n9^n \\
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^n \cdot \left( 10 - (n+1) \left( \frac{9}{10} \right)^n - 10 \left( \frac{9}{10} \right)^{n+1} \right) \right) + n9^n \\
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^n \cdot \left( 10 - (n+1) \left( \frac{9}{10} \right)^n - 9 \left( \frac{9}{10} \right)^n \right) \right) + n9^n \\
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^n \cdot \left( 10 - (n+10) \left( \frac{9}{10} \right)^n \right) \right) + n9^n \\
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^n \cdot \left( 10 - n \left( \frac{9}{10} \right)^n + 10 \left( \frac{9}{10} \right)^n \right) \right) + n9^n \\
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10^n \cdot \left( \frac{10(10^n) - n(9^n) + 10(9^n)}{10^n} \right) \right) + n9^n \\
Avg_{Search}(n) &= \frac{1}{10^n} \cdot \left( \sum_{x=1}^{10} 10(10^n) - n(9^n) + 10(9^n) + n9^n \right) \\
Avg_{Search}(n) &= \sum_{x=1}^{10} \frac{10(10^n) + 10(9^n)}{10^n} \\
Avg_{Search}(n) &= \sum_{x=1}^{10} 10 + 10 \left( \frac{9}{10} \right)^n
\end{aligned}$$

Since  $\frac{9}{10} < 1$ ,  $\left( \frac{9}{10} \right)^n$  approaches 0 as  $n$  increases. Ignoring this term as it approaches 0, we have the average run time is equal to a constant value, 100, which is independent of the length of the input size. Then by definition of Big-Theta,  $Avg_{Search}(n) \in \theta(1)$ . ■

b) The proof of exercise 5.4 using the values 1-10 is analogous to using 1-500, as in both cases the upper limit of values is not dependent on the size of the input list. Intuitively, the proof for values of 1-500, would be the same as above except with every 10 replaced with 500, and every 9 replaced with 499. Then the average runtime would be  $500^2$ , which is still in  $\Theta(1)$ . ■

c) Define the input size of counter as  $n = s + u$ . We want to show the  $WC - RT_{counter}(n) \in \Theta(n)$ .

**Proof:**

We want to show that if 7 iterations occur, then  $n$  decreases by at least 1.

Case 1: Assume  $u \neq 0$ , then by the pre-condition,  $1 \leq u \leq 6$ .

In this case,  $n = s + u$ . Since  $u$  is not equal to 0, the if condition will not execute, and the else condition



will execute, subtracting one from  $u$ . This will iterate at most 6 times as  $u \leq 6$  and when  $u = 0$  the if condition will be true. Once we reach this state,  $n = s$  (because  $u = 0$ ), and therefore  $n$  has been decreased by atleast one in less than 7 iterations.

Case 2: Assume  $u = 0$ .

In this case,  $n = s$ , (because  $u = 0$ ). Then the if statement will evaluate true and execute, setting  $u$  equal to 6, and  $s$  one less. On the next iteration, the if statement will evaluate false, and the else will execute, setting  $u$  to one less. The else will execute for 6 iterations until  $u$  is equal to 0 again. Then after these 7 iterations,  $s$  has been decreased by 1, and  $u$  is still equal to 0, therefore  $n$  has decreased by one.

In both cases, we've shown  $n$  is decreased by one after at most 7 iterations. Notice that Case 1 can only occur once at the start, as the result in both cases is where  $u = 0$ , which is the condition for Case 2.

For all  $k \in \mathbb{N}$ , define  $n_k$  as the value of  $n$  after  $7k$  iterations, in the case that  $7k$  iterations occur. For any value of  $k$ , either the loop terminates within  $7k$  iterations, or the value of  $n$  has decreased by  $k$ .

Then  $n_k = n - k$ , and the loop terminates when  $n_k \leq 0$ .

$$\begin{aligned} n_k &= n - k \leq 0 \\ n - k &\leq 0 \\ n &\leq k \end{aligned}$$

So  $n_k$  is less than or equal to zero when  $k$  is greater than or equal to  $n$ . Then  $WC - RT_{counter}(n)$  will have at most  $7n$  steps, which is in  $O(n)$ .

Now we want to show there exists an input family such that  $WC - RT_{counter}(n) \in \Omega(n)$

Let  $n \in \mathbb{N}$ .

Take  $u = 0$

Take  $s = n$

Then this input satisfies the conditions of case 1, where  $u = 0$ . By case 1, after 7 iterations,  $n$  will decrease by 1, and  $u$  will still equal 0. Therefore this family of inputs will have  $7n$  steps. Therefore there exists an input family such that  $WC - RT_{counter}(n) \in \Omega(n)$ . Therefore  $WC - RT_{counter}(n) \in \Theta(n)$  as needed. ■

d) **Proof:**

$s + u$  reduces by a maximum of 1 every iteration of the while loop, so the loop will execute a minimum of  $s + u = n$  times. Both lines 4 and 12 only execute once, so in the best case, the program will take a maximum of  $n + 2$  steps. So,

$$RT(BC(counter)) \in \Omega(n)$$

When  $u = 6$  and  $s = n - 6$ ,

Every 7 iterations of the while loop, after the else block will execute, and  $s$  will be lowered by one. This will continue until  $s = 0$ .

Lines 4 and 12 each take 1 step to execute.

So, the minimum total number of steps is

$$7(n - 6) + 2$$

So,

$$RT(BC(counter)) \in O(n)$$

$$\text{Since } RT(BC(counter)) \in \Omega(n) \wedge RT(BC(counter)) \in O(n),$$

$$RT(BC(counter)) \in \Theta(n)$$

■

e) The average runtime of a program must be between its worst-case and best-case runtimes, otherwise one of the asymptotic bounds would be incorrect. For example, if you had a worst case running time in  $\Theta(n)$ , and an average running time in  $\Theta(n^2)$ , one of these bounds would be incorrect since this would imply the worst case running time is in fact at least  $\Theta(n^2)$ .

In this case, the average running time of *counter* must be in between both the worst-case and best-case running times of *counter*, which are both in  $\Theta(n)$ . Therefore the average running time of *counter* is in  $\Theta(n)$ .