Csc236

1.

(a)

The number of elements of T that have

0 left parentheses: 1

*

1 left parentheses: 1

(**)

2 left parentheses: 2

(*(**)), ((**)*)

3 left parentheses: 5

((*(**))*), (((**)*)*),

((**)(**)),

(*(*(**))), (*((**)*))

4 left parentheses: 14

(((*(**))*)*), ((((**)*)*)*), (((**)(**))*), ((*(*(**)))*), ((*((**)*))*),

((*(**))(**)), (((**)*)(**)),

((**)(*(**))), ((**)((**)*)),

(*((*(**))*)), (*(((**)*)*)), (*((**)(**))), (*(*(*(**)))), (*(*((**)*)))

(b)

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

Define $\mathrm{c(n)}$: the number of different elements of T with n left parentheses.

Then

$$\mathrm{c(n)} = \begin{cases} \sum_{i=1}^{n} c(n-i) \cdot c(i-1) & if\ n > 0 \\ 1 & if\ n = 0 \end{cases}$$

Explanation:

<u>Case $n = 0$</u>:

From 1(a) we know that there is only 1 element in T that has 0 left parentheses.

So $\mathrm{c(n)} = 1$

Case $n > 0$:

Let $x_1, x_2 \in T$.

Let $a_1, a_2 \in \mathbb{N}$ be the number of left parentheses x_1, x_2 has, respectively.

For any combination of x_1, x_2: $(x_1 x_2)$ has $a_1 + a_2 + 1$ left parentheses.

Let x be an arbitrary element of T with n left parentheses.

Since $n > 0$, x must be a combination of two elements from T.

Let x_1, x_2 be such two elements.

Then x = $(x_1 x_2)$ with $a_1 + a_2 + 1 = \mathrm{n}$ left parentheses.

Thus $a_1 + a_2 = n - 1$

Hence, we only need to figure out how many different combinations of x_1, x_2 with $a_1 + a_2 = n - 1$ can make $(x_1 x_2)$ have n left parentheses.

We know that:

$$n - 1 = (n - 1) + (0) = (n - 2) + (1) = (n - 3) + (2) = \cdots = (0) + (n - 1)$$

Since $\mathrm{n} \in \mathbb{N}$, all of $\mathrm{c}(\mathrm{n} - 1), c(n - 2), \ldots, c(0)$ can be defined.

Also, different x_1, x_2 can make different $(x_1 x_2)$. There are $\mathrm{c}(a_1) \cdot c(a_2)$ different ways to form $(x_1 x_2)$ when a_1, a_2 are fixed to be specific natural numbers such that $a_1 + a_2 = n - 1$.

Hence:

$$\begin{aligned}\mathrm{c}(\mathrm{n}) &= c(n-1) \cdot c(0) + c(n-2) \cdot c(1) + c(n-3) \cdot c(2) + \cdots + c(0) \cdot c(n-1) \\ &= \sum_{i=1}^{n} c(n-i) \cdot c(i-1)\end{aligned}$$

When n is even:

The middle two terms of $\sum_{i=1}^{n} c(n-i)\cdot c(i-1)$ are

$$c\left(\frac{n}{2}\right)\cdot c\left(\frac{n-1}{2}\right)+c\left(\frac{n-1}{2}\right)\cdot c\left(\frac{n}{2}\right)$$

These are two different combinations.

Thus, all the terms are different.

So, there is no repeat.

When n is odd:

The middle term of $\sum_{i=1}^{n} c(n-i)\cdot c(i-1)$ is

$$c\left(\frac{n-1}{2}\right)\cdot c\left(\frac{n-1}{2}\right)$$

This term only appears once.

Thus, all the terms are different.

So, there is no repeat.

Hence, we can say that there is no repeat.

Therefore, I've explained why

$$\mathrm{c(n)}=\begin{cases}\displaystyle\sum_{i=1}^{n} c(n-i)\cdot c(i-1) & if\ n>0\\ 1 & if\ n=0\end{cases}$$

2.

(a)

Answer:

$$\mathrm{p(n)}=\begin{cases}\begin{cases}0 & if\ n=1\ or\ 2\\ 1 & if\ n=0,3,4,5,6,7\\ 2 & if\ n=8,9,10,11\\ 3 & if\ n=12\end{cases}\\ p(n-3)+p(n-4)+p(n-5)-p(n-7)-p(n-8)-p(n-9)+p(n-12) & if\ n>12.\end{cases}$$

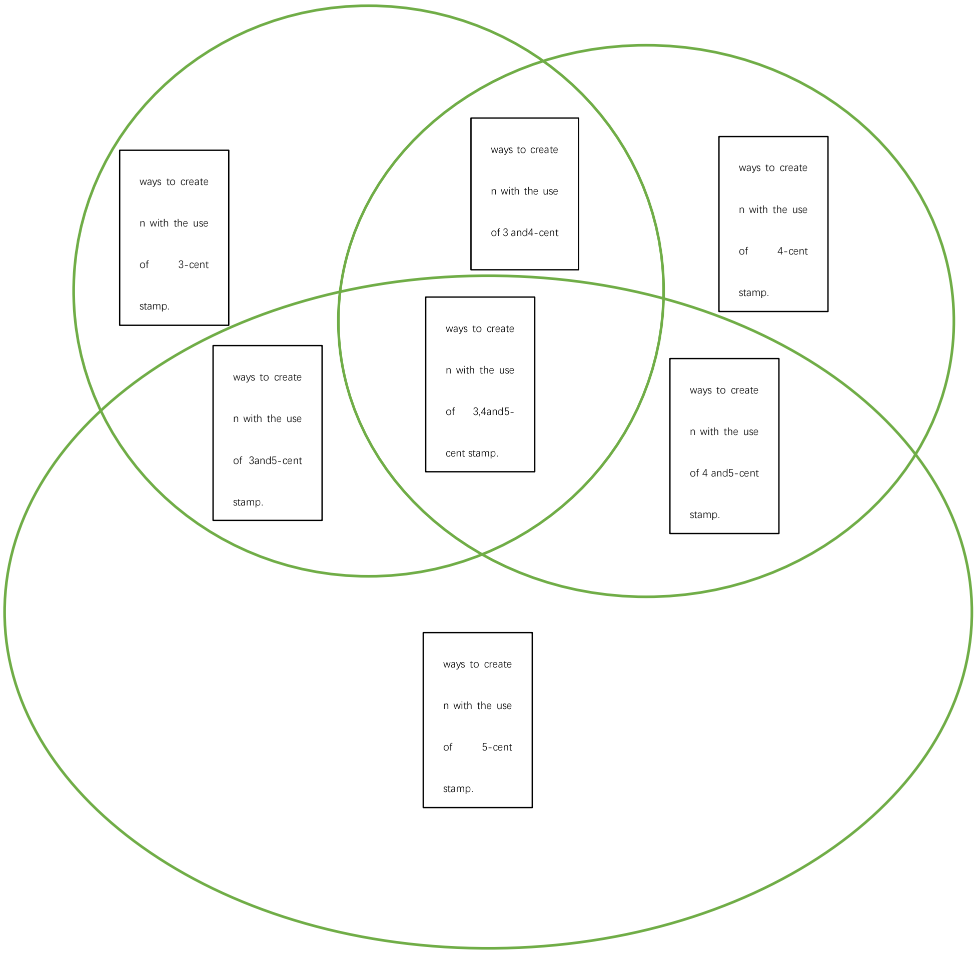
The explanation is the following:

Since $0=0\times 3+0\times 4+0\times 5$, hence p(0)=1,

Since 0<1<2<3<4<5, hence p(1)=p(2)=0,

Since $3=3\times 1, 4=4\times 1, 5=5\times 1$, hence p(3)=p(4)=p(5)=1,

Since $6 = 3 + 3, 7 = 3 + 4$, hence p(6)=p(7)=1,
Since $8 = 4 + 4 = 5 + 3, 9 = 3 \times 3 = 4 + 5, 10 = 5 + 5 = 3 \times 2 + 4, 11 = 5 + 3 \times 2 = 4 \times 2 + 3$, hence p(8)=p(9) =p(10)=p(11)=2,
Since $12 = 3 \times 4 = 4 \times 3 = 3 + 4 + 5$, hence p(12)=3.
To explain the function for n>12, I will first explain that p(n-3)=#of ways to create n with the use of 3-cent stamp: since p(n-3)= #of ways to create n-3, let x be an arbitrary way of the ways to create n-3, then the combination of x and a 3-cent stamp must be a way to create n, hence (x+ 3-cent stamp) is a way of the ways to create n with the use of 3-cent stamp. And different x has different (x+ 3-cent stamp). Hence p(n-3)≤ #of ways to create n with the use of 3-cent stamp. Let y be an arbitrary way of the ways to create n with the use of 3-cent stamp. Then if we take away a 3-cent stamp form the combination of y, the remaining combination will be a way of the ways to create n-3. hence (y- 3-cent stamp) is a way of the ways to create n-3. And different y has different (y- 3-cent stamp). Hence #of ways to create n with the use of 3-cent stamp ≤p(n-3). Hence we can conclude that p(n-3) = #of ways to create n with the use of 3-cent stamp.
And just like the above, p(n-4) = #of ways to create n with the use of 4-cent stamp,
p(n-5) = #of ways to create n with the use of 5-cent stamp,
p(n-7) = #of ways to create n with the use of 3-cent stamp and 4-cent stamp,
p(n-8) = #of ways to create n with the use of 3-cent stamp and 5-cent stamp,
p(n-9) = #of ways to create n with the use of 5-cent stamp and 4-cent stamp,
p(n-12) = #of ways to create n with the use of 3-cent stamp and 4-cent stamp and 5-cent stamp.
Since n>12, all of p(n-4), p(n-5), p(n-7), p(n-8), p(n-9), p(n-12) make sense.
And we know that
#of ways to create n = #of ways to create n with the use of 3-cent stamp + #of ways to create n with the use of 4-cent stamp+#of ways to create n with the use of 5-cent stamp-#of ways to create n with the use of 3-cent stamp and 4-cent stamp-#of ways to create n with the use of 3-cent stamp and 5-cent stamp-#of ways to create n with the use of 5-cent stamp and 4-cent stamp+#of ways to create n with the use of 3-cent stamp and 4-cent stamp and 5-cent stamp.



(we use a picture to explain this)

Hence when n>12, $p(n) = p(n-3) + p(n-4) + p(n-5) - p(n-7) - p(n-8) - p(n-9) + p(n-12)$.

Hence

$$p(n) = \begin{cases} \begin{cases} 0 & if\ n = 1\ or\ 2 \\ 1 & if\ n = 0,3,4,5,6,7 \\ 2 & if\ n = 8,9,10,11 \\ 3 & if\ n = 12 \end{cases} \\ p(n-3) + p(n-4) + p(n-5) - p(n-7) - p(n-8) - p(n-9) + p(n-12) & if\ n > 12. \end{cases}$$

(b)

WTS: $\forall n \in \mathbb{N}^+$, p(n) is monotonic nondecreasing

Proof:

We are going to prove that $\forall n \in \mathbb{N}^+$, p(n) is monotonic nondecreasing. That is to prove that $\forall n \in \mathbb{N}^+, n > 1 \Rightarrow$ p(n) ≥ p(n-1). That is $\forall n \in \mathbb{N}^+, n > 1 \Rightarrow$ p(n)-p(n-1) ≥ 0.

Define q(n) := p(n)-p(n-1) ≥ 0. So we are going to prove that $\forall n \in \mathbb{N}^+, n > 1 \Rightarrow$ q(n)

I am going to prove this by complete induction.

Inductive step:

Let $n \in \mathbb{N}^+, assume\ n >$
$1, assume\ H(n): \Lambda_{i=2}^{i=n-1} q(i). we\ will\ prove\ q(n)\ follows, that\ is\ p(n) - p(n-1) \geq 0$.

- Base case n=2: p(2)-p(1)=0-0=0≥ 0, so q(n) follows in this case.

- Base case n=3: p(3)-p(2)=1-0=1≥ 0, so q(n) follows in this case.
- Base case n=4: p(4)-p(3)=1-1=0≥ 0, so q(n) follows in this case.
- Base case n=5: p(5)-p(4)=1-1=0≥ 0, so q(n) follows in this case.
- Base case n=6: p(6)-p(5)=1-1=0≥ 0, so q(n) follows in this case.
- Base case n=7: p(7)-p(6)=1-1=0≥ 0, so q(n) follows in this case.
- Base case n=8: p(8)-p(7)=2-1=1≥ 0, so q(n) follows in this case.
- Base case n=9: p(9)-p(8)=2-2=0≥ 0, so q(n) follows in this case.
- Base case n=10: p(10)-p(9)=2-2=0≥ 0, so q(n) follows in this case.
- Base case n=11: p(11)-p(10)=2-2=0≥ 0, so q(n) follows in this case.
- Base case n=12: p(12)-p(11)=3-2=1≥ 0, so q(n) follows in this case.
- Base case n=13: p(13)-p(12)=3-3=0≥ 0, so q(n) follows in this case.
- Base case n=14: p(14)-p(13)=3-3=0≥ 0, so q(n) follows in this case.
- Base case n=15: p(15)-p(14)=4-3=1≥ 0, so q(n) follows in this case.
- Case n>15:

  Since n>15, n-1>14, according to the definition of the function,

$$
\begin{aligned}
\mathrm{p(n)} - \mathrm{p(n-1)} &= p(n-3) + p(n-4) + p(n-5) - p(n-7) - p(n-8) - p(n-9) \\
&\quad + p(n-12) - p(n-4) - p(n-5) - p(n-6) + p(n-8) + \\
&\quad p(n-9) + p(n-10) - p(n-13) \\
&= p(n-3) - p(n-7) + p(n-12) - p(n-6) + p(n-10) - \\
&\quad p(n-13) \\
&= p(n-3) - p(n-6) - p(n-7) + p(n-10) + p(n-12) - \\
&\quad p(n-13)
\end{aligned}
$$

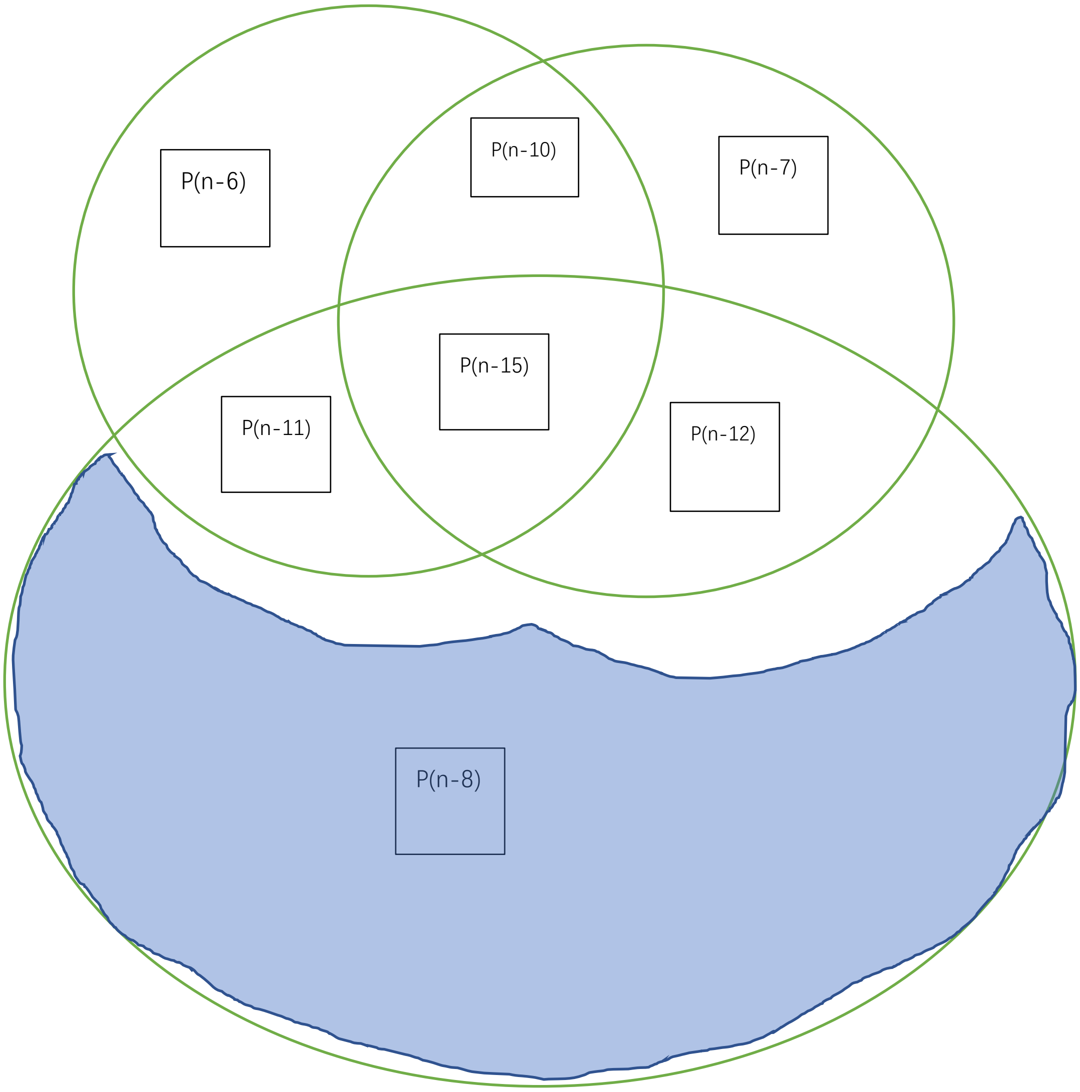
  Since n>15, 2<n-12<n, hence by inductive hypothesis we know that q(n-12) is true, that is p(n-12)-p(n-13)≥ 0

  Hence, we only need to show that $p(n-3) - p(n-6) - p(n-7) + p(n-10) \geq 0$

  Since n>15, n-3>12, according to the definition of the function,

$$
\begin{aligned}
p(n-3) &= p(n-6) + p(n-7) + p(n-8) - p(n-10) - p(n-11) - p(n-12) \\
&\quad + p(n-15)
\end{aligned}
$$

  hence $p(n-3) - p(n-6) - p(n-7) + p(n-10) = p(n-8) - p(n-11) - p(n-12) + p(n-15)$, and I can prove that $p(n-8) - p(n-11) - p(n-12) + p(n-15) \geq 0$ with the following picture:



We can see that $p(n-8)-p(n-11)-p(n-12)+p(n-15)$ is actually the blue space, with $p(n-6)+p(n-7)-p(n-10)$ be the white space, and the combination of them is p(n-3).

Since the blue space must have non-negative ways hence $p(n-8)-p(n-11)-p(n-12)+p(n-15)$ must be non-negative, that is $p(n-8)-p(n-11)-p(n-12)+p(n-15)\geq 0$ That is $p(n-3)-p(n-6)-p(n-7)+p(n-10)\geq 0$

Hence $p(n-3)-p(n-6)-p(n-7)+p(n-10)+p(n-12)-p(n-13)\geq 0$

So q(n) follows in this case.

Hence we have proved that $\forall \mathrm{n} \in \mathbb{N}^{+}$, p(n) is monotonic nondecreasing.■

3.

(a)

Proof:

Define

$$\mathrm{P(n)}: \bigwedge_{m=1}^{m=n} T(m) \leq T(n)$$

I will prove that $\forall \mathrm{n} \in \mathbb{N}^{+},\ P(n)$ using complete induction.

I will assume, without proof, that for any integer $\mathrm{n} > 1, \mathrm{n} > \left\lceil \frac{n}{2} \right\rceil \geq 1$. (by lemma 3.3 from the course notes)

<u>Inductive step:</u>

Let $\mathrm{n} \in \mathbb{N}^{+}$, assume $\bigwedge_{i=1}^{i=n-1} P(i)$.

I will show that $P(n)$ follows.

<u>Base case $\mathrm{n} \leq 2$:</u>

$\mathrm{T}(1) = c^{'}$, which is no smaller than the value of T on any smaller positive natural number, since there are no smaller natural numbers. So $P(1)$ holds.

$\mathrm{T}(2) = 1 + \mathrm{T}(1) = 1 + c^{'} \geq c^{'} = T(1)$, so $P(2)$ holds since 1 is the only positive natural number less than 2.

Case $\mathrm{n} \geq 3$:

Since $n > 2,\ n > n-1 > 1$.

So $P(n-1)$ holds (by inductive hypothesis) and (by transitivity of $\leq$) we need only show that $T(n-1) \leq T(n)$.

Since $n-1 > 1$ we have:

$$T(n-1) = 1 + T(\lceil \tfrac{n-1}{2} \rceil) \qquad \text{\# by definition of T, since } n-1 > 1$$

$$\leq 1 + T(\lceil \tfrac{n}{2} \rceil) \qquad \text{\# by } P(\lceil \tfrac{n}{2} \rceil) \text{ since } \mathrm{n} > \lceil \tfrac{n}{2} \rceil \geq \lceil \tfrac{n-1}{2} \rceil \geq 1$$

$$= T(n)$$

From the above, I've proven that $\forall \mathrm{n} \in \mathbb{N}^{+},\ P(n)$.

Therefore, T is nondecreasing.

■

(b)

Proof:

Define

$$\mathrm{P(k)}: \mathrm{T}(2^k) = k + c^{'}$$

Note that when $n \in \mathbb{N}$ and $n = 2^k$, this is equivalent to $\mathrm{T(n)} = \lg(\mathrm{n}) + c^{'}$

I will prove that $\forall \mathrm{k} \in \mathbb{N},\ \mathrm{P(k)}$ using simple induction on k.

<u>Base case:</u>

$\mathrm{T}(2^0) = T(1) = c^{'} = 0 + c^{'}$

So $\mathrm{P}(0)$ holds

<u>Inductive step:</u>

Let $\mathrm{k} \in \mathbb{N}$

Assume $\mathrm{P(k)}$, that is $\mathrm{T}(2^k) = k + c^{'}$.

I will show that $\mathrm{P(k+1)}$ follows, that is $\mathrm{T}(2^{k+1}) = k + 1 + c^{'}$.

Since $\mathrm{k} + 1 > 0,\ 2^{k+1} > 1$, so by definition of T:

$$T(2^{k+1}) = 1 + T\left(\left\lceil \frac{2^{k+1}}{2} \right\rceil\right) \quad \text{\# by definition of T, since } 2^{k+1} > 1$$
$$= 1 + T(\lceil 2^k \rceil)$$
$$= 1 + T(2^k) \quad \text{\# since } 2^k \text{ is an integer}$$
$$= 1 + k + c^{'} \quad \text{\# by inductive hypothesis, } \mathrm{P(k)} \text{ holds, that is } \mathrm{T}(2^k) = k + c^{'}$$

From the above, I've proven that $\forall \mathrm{k} \in \mathbb{N},\ \mathrm{P(k)}$.

Therefore, $\forall k, n \in \mathbb{N}, n = 2^k \Rightarrow T(n) = \lg(n) + c^{'}$.

■

(c)

Proof:

Wants to show: $T \in \Theta(\lg(n))$

Define $n^* = 2^{\lceil \lg(n) \rceil}$

Then

$$\lceil \lg(n) \rceil - 1 < \lg(n) \le \lceil \lg(n) \rceil \Rightarrow \frac{n^*}{2} < n \le n^*$$

From part (a) we know: T is nondecreasing

From part (b) we know: $\forall \mathrm{k} \in \mathbb{N},\ \mathrm{T}(2^k) = k + c^{'}$

Show that $T \in \mathrm{O}(\lg(n))$:

Let $\mathrm{d} = 2$. Then $\mathrm{d} \in \mathbb{R}^+$.

Let $\mathrm{B} = 2$. Then $\mathrm{B} \in \mathbb{N}^+$.

Let n be an arbitrary natural number no smaller than B.

Then

$$\begin{aligned} T(n) &\le T(n^*) \quad \text{\# since T is nondecreasing and } n \le n^* \\ &= \lg(n^*) + c^{'} \quad \text{\# since } \forall \mathrm{k} \in \mathbb{N},\ \mathrm{T}(2^k) = k + c^{'} \\ &< \lg(2n) + c^{'} \quad \# \tfrac{n^*}{2} < n \Rightarrow n^* < 2n \\ &= \lg(n) + 1 + c^{'} \\ &\le \lg(n) + \lg(n) + c^{'} \quad \# n \ge B = 2 \Rightarrow \lg(n) \ge 1 \\ &= 2\lg(n) + c^{'} \\ &= d\lg(n) + c^{'} \quad \text{\# since d} = 2 \end{aligned}$$

Since $c^{'}$ is the time cost for some operations, $c^{'}$ must be a constant no smaller than 0.

Therefore, $T \in \mathrm{O}(\lg(n))$ since $d\lg(n) + c^{'}$ differs from $d\lg(n)$ by adding a constant $c^{'}$.

Show that $T \in \Omega(\lg(n))$:

Let $\mathrm{d} = \frac{1}{2}$. Then $\mathrm{d} \in \mathbb{R}^{+}$.

Let $\mathrm{B} = 4$. Then $\mathrm{B} \in \mathbb{N}^{+}$.

Let n be an arbitrary natural number no smaller than B.

Then

$$T(n) \geq T(\frac{n^*}{2}) \quad \text{\# since T is nondecreasing and } n > \frac{n^*}{2}$$

$$= \lg\left(\frac{n^*}{2}\right) + c' \quad \text{\# since } \forall \mathrm{k} \in \mathbb{N},\ \mathrm{T}(2^k) = k + c'$$

$$\geq \lg\left(\frac{n}{2}\right) + c' \quad \# \, n^* \geq n \Rightarrow \frac{n^*}{2} \geq \frac{n}{2}$$

$$= \lg(n) - 1 + c'$$

$$= \frac{1}{2}\lg(n) + \frac{1}{2}\lg(n) - 1 + c'$$

$$\geq \frac{1}{2}\lg(n) + c' \quad \# \, n \geq B = 4 \Rightarrow \frac{1}{2}\lg(n) \geq 1$$

$$= d\lg(n) + c' \quad \text{\# since } \mathrm{d} = \frac{1}{2}$$

Since c' is the time cost for some operations, c' must be a constant no smaller than 0.

Therefore, $T \in \Omega(\lg(n))$ since $d\lg(n) + c'$ differs from $d\lg(n)$ by adding a constant c'.

From the above, I've proven that $T \in \mathrm{O}(\lg(n))$ and $T \in \Omega(\lg(n))$.

Therefore, $T \in \Theta(\lg(n))$.

■

4.

(a) I will prove that its precondition plus execution implies its postcondition. The proof is the following:

Proof:

Define $\mathrm{c(j)}: 0 \leq \mathrm{i} \leq \mathrm{len(s1)}$ and $\mathrm{j} \leq \mathrm{len(s2)}$

$\Longrightarrow$ returns numbers of times s1[: i] occurs as a subsequence of s2[: j]

I will use complete induction to prove $\forall \mathrm{j} \in \mathbb{N}, \mathrm{c(j)}$.

Inductive step:

Let $\mathrm{j} \in \mathbb{N}$, $assume\ H(j): \Lambda_{i=0}^{i=n-1} c(i). we\ will\ prove\ c(j)\ follows, assume\ 0 \leq \mathrm{i} \leq$

$\mathrm{len(s1)\ and\ j} \leq$
$\mathrm{len(s2)\ and\ we\ will\ prove\ the\ code\ returns\ numbers\ of\ times\ s1[:i]\ occurs\ as\ a\ subsequence\ of\ s2[:j]}$
.

- Base case: j=0, and we can divide this situation in to 2 cases

1. i=0=j, then according to the code, it will return 1, since $\mathrm{numbers\ of\ times\ ""\ occurs\ as\ a\ subsequence\ of\ ""}$ is 1, hence the result is right.
2. i>0=j, then according to the code, it will return 0, since $\mathrm{numbers\ of\ times\ "\ldots"\ occurs\ as\ a\ subsequence\ of\ ""}$ is 0, hence the result is right.

   So c(j) follows in this case

- Case: j>0, and we can divide this situation in to 4 cases.
  1. i=0, then according to the code, it will return 1, since $\mathrm{numbers\ of\ times\ ""\ occurs\ as\ a\ subsequence\ of\ "\ldots"}$ is 1, hence the result is right.
  2. i>j, then according to the code, it will return 0, since $\mathrm{numbers\ of\ times\ "\ldots\ldots"(larger\ length)\ occurs\ as\ a\ subsequence\ of\ "\ldots"(smaller\ length)}$ is 0, hence the result is right.
  3. $0 < \mathrm{i} \leq \mathrm{j\ and\ s1[i-1]} \neq \mathrm{s2[j-1]}$, then according to the code, it will return count_subsequence(s1,s2,i,j-1), since j>0, $0 \leq \mathrm{j} - 1 < \mathrm{j}$, hence according to the inducive hypothesis, since $0 \leq \mathrm{i} \leq \mathrm{len(s1)\ and\ j} \leq \mathrm{len(s2)}$, hence $0 \leq \mathrm{i} \leq \mathrm{len(s1)\ and\ j} - 1 \leq \mathrm{len(s2)}$, hence $\mathrm{the\ code\ returns\ numbers\ of\ times\ s1[:i]\ occurs\ as\ a\ subsequence\ of\ s2[:j-1]}$. Since $\mathrm{s1[i-1]} \neq \mathrm{s2[j-1]}$ and s2[j-1] is the last term of s2[:j], s1[i-1] is the last term of s1[:i],hence any subsequence that use s2[j-1] (if it uses then s2[j-1] must be the last term) will not be $\mathrm{s1[:i]}$, in other word, we only need to consider about $\mathrm{s2[:j-1]}$, hence the $\mathrm{numbers\ of\ times\ s1[:i]\ occurs\ as\ a\ subsequence\ of\ s2[:j-1]}$ is equal to the $\mathrm{numbers\ of\ times\ s1[:i]\ occurs\ as\ a\ subsequence\ of\ s2[:j]}$, hence the result is right.
  4. $0 < \mathrm{i} \leq \mathrm{j\ and\ s1[i-1]} = \mathrm{s2[j-1]}$, then according to the code, it will return count_subsequence(s1,s2,i,j-1)+ count_subsequence(s1,s2,i-1,j-1), since j>0, $0 \leq \mathrm{j} - 1 < \mathrm{j}$, hence according to the inducive hypothesis, since $0 < \mathrm{i} \leq \mathrm{len(s1)\ and\ j} \leq \mathrm{len(s2)}$, hence $0 \leq \mathrm{i} \leq \mathrm{len(s1)\ and\ j} - 1 \leq \mathrm{len(s2)and}\ 0 \leq \mathrm{i} - 1 \leq \mathrm{len(s1)}$, hence $\mathrm{the\ code\ \ returns\ numbers\ of\ times\ s1[:i]occurs\ as\ a\ subsequence\ ofs2[:j-1]} +$ $\mathrm{numbers\ of\ times\ s1[:i-1]occurs\ as\ a\ subsequence\ of\ s2[:j-1]}$. Since $\mathrm{s1[i-1]} = \mathrm{s2[j-1]}$ and s2[j-1] is the last term of s2[:j], s1[i-1] is the last term of s1[:i], hence we can consider the subsequence into 2 situation. One situation is I will use $s2[j-1]$ as a term of the subsequence, since the last term of the sequence has been defined and is of course equal to the last term of $s1[:\mathrm{i}]$, hence we only need to consider the numbers of $times\ s1[:\mathrm{i}-1]occurs\ as\ a\ subsequence\ of\ s2[:\mathrm{j}-1]$. Hence in this situation the number is equal to the numbers of $times\ s1[:\mathrm{i}-1]occurs\ as\ a\ subsequence\ of\ s2[:\mathrm{j}-1]$. In the other situation, I will not use $s2[j-1]$ as a term of the subsequence, hence we only need to consider about the numbers of $times\ s1[:\mathrm{i}]occurs\ as\ a\ subsequence\ of s2[:\mathrm{j}-1]$, hence in this situation the number is equal to the numbers of $times\ s1[:\mathrm{i}]occurs\ as\ a\ subsequence\ of\ s2[:\mathrm{j}-1]$. So we combinate the 2 situation together and the total number is

$numbers\ of\ times\ s1[:\mathrm{i}]occurs\ as\ a\ subsequence\ of s2[:\mathrm{j}-1]+$
$numbers\ of\ times\ s1[:i-1]occurs\ as\ a\ subsequence\ of\ s2[:j-1]$
hence the result is right.

So c(j) follows in this case

Hence we have proved that $\forall \mathrm{j} \in \mathbb{N}, \mathrm{c(j)}$, that is its precondition plus execution implies its postcondition. ■

5.

I am going to show that my function is right, the proof is the following:

Proof:

Assume the precondition that is colour_list is a List[str] from {"b","g","r"}.

Let $blue_i$ $be\ blue\ after\ the\ ith\ iteration.$ $Let\ green_i\ be\ green\ after\ the\ ith\ iteration.$ $Let\ red_i\ be\ red\ after\ the\ ith\ iteration.$

Define p(i): after the ith iteration of the loop (if it occurs), $0 \leq blue_i \leq green_i \leq red_i \leq len(colour\_list)$ and $\text{colour\_list}[0{:}\,green_i] + colour\_list[red_i{:}\,]\ same\ colours\ as\ before$ and $\text{all([c == "b" for c in colour\_list}[0{:}\,blue_i]])$ and $\text{all([c == "g" for c in colour\_list}[blue_i{:}\,green_i]])$ and $\text{all([c == "r" for c in colour\_list}[red_i{:}\,]])$

I am going to prove $\forall \mathrm{i} \in \mathbb{N}, \mathrm{p(i)}$ using simple induction on i.

Base case: $blue_0 = 0,\ green_0 = 0,\ red_0 = len(colour\_list)$ (by initialization),

Since $0 \leq 0 \leq 0 \leq len(colour\_list) \leq len(colour\_list)$, hence $0 \leq blue_0 \leq green_0 \leq red_0 \leq len(colour\_list).$

$\text{colour\_list}[0{:}\,green_0] + colour\_list[red_0{:}\,] = \text{colour\_list}[0{:}\,0] + colour\_list[len(colour\_list){:}\,] = []$ and it has the same colours as before since it is the 0th iteration.

$\text{all([c == "b" for c in colour\_list}[0{:}\,blue_0]]) = \text{all([c == "b" for c in colour\_list}[0{:}\,0]]) = \text{all([c == "b" for c in []])}$ must always be true.

$\text{all([c == "g" for c in colour\_list}[blue_0{:}\,green_0]]) = \text{all([c == "g" for c in colour\_list}[0{:}\,0]]) = \text{all([c == "g" for c in []])}$ must always be true.

$\text{all([c == "r" for c in colour\_list}[red_0{:}\,]]) = \text{all([c == "r" for c in colour\_list}[len(colour\_list){:}\,]]) = \text{all([c == "r" for c in []])}$ must always be true.

So p(0) follows.

Inductive step:

Let $\mathrm{i} \in \mathbb{N}$ and assume p(i). Show that p(i+1) follows.(If there is an (i+1)th loop iteration)

Since i+1>0 hence we will excute the while loop.

According to the code I will divide the proof into 7 cases.

- Case1: colour_list[red_i - 1] = "r" and colour_list[$green_i$] = "g".

According to the code $red_{i+1} = red_i - 1$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i$.

Since by inductive hypothesis we know that $0 \leq blue_i \leq green_i \leq red_i \leq len(colour\_list)$

And since we have the i+1 iteration, $green_i < red_i$

Hence $0 \leq blue_i \leq green_i < red_i \leq len(colour\_list)$

Hence $red_{i+1} < red_i \leq len(colour\_list)$, that is $red_{i+1} \leq len(colour\_list)$

Also $0 \leq blue_i = blue_{i+1} \leq green_i < green_{i+1}$, hence $0 \leq blue_{i+1} \leq green_{i+1}$

And since we have colour_list[red_i - 1] = "r" and colour_list[$green_i$] = "g" and they are not equal to each other, hence $red_i - 1 > green_i$ that is $red_i - 1 \geq green_i + 1$ that is $red_{i+1} \geq green_{i+1}$

Combining the above, we have $0 \leq blue_{i+1} \leq green_{i+1} \leq red_{i+1} \leq len(colour\_list)$

Since by inductive hypothesis we know that colour_list$[0{:}green_i] + colour_list[red_i{:}]$ $same\ colours\ as\ before$ .Since we didn't change any element of the list in this case, colour_list$[0{:}green_{i+1}] + colour_list[red_{i+1}{:}]$ $same\ colours\ as\ last\ iteration\ as\ before$

Since by inductive hypothesis we know that all([c == "b" for c in colour_list[0: $blue_i$]]) and $blue_{i+1} = blue_i$ hence all([c == "b" for c in colour_list[0: $blue_{i+1}$]])

Since by inductive hypothesis we know that all([c == "g" for c in colour_list[$blue_i$: $green_i$]]) and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i$ and colour_list[$green_i$] = "g", hence all([c == "g" for c in colour_list[$blue_{i+1}$: $green_{i+1}$]])

Since by inductive hypothesis we know that all([c == "r" for c in colour_list[red_i:]]) and $red_{i+1} = red_i - 1$ and colour_list[red_i - 1] = "r", hence all([c == "r" for c in colour_list[red_{i+1}:]]).

So p(i+1) follows in this case.

- Case2: colour_list[red_i - 1] = "r" and colour_list[$green_i$] = "r".

According to the code $red_{i+1} = red_i - 1$ and $green_{i+1} = green_i$ and $blue_{i+1} = blue_i$.

Since by inductive hypothesis we know that $0 \leq blue_i \leq green_i \leq red_i \leq len(colour\_list)$

And since we have the i+1 iteration, $green_i < red_i$

Hence $0 \leq blue_i \leq green_i < red_i \leq len(colour\_list)$

Hence $red_{i+1} < red_i \leq len(colour\_list)$, that is $red_{i+1} \leq len(colour\_list)$

Also $0 \leq blue_i = blue_{i+1} \leq green_i = green_{i+1}$, hence $0 \leq blue_{i+1} \leq green_{i+1}$

And since $red_i > green_i$ that is $red_i - 1 \geq green_i$ that is $red_{i+1} \geq green_{i+1}$

Combining the above, we have $0 \leq blue_{i+1} \leq green_{i+1} \leq red_{i+1} \leq len(colour\_list)$

Since by inductive hypothesis we know that colour_list$[0{:}green_i] + colour_list[red_i{:}]$ $same\ colours\ as\ before$ .Since we didn't change any element of the list in this case, colour_list$[0{:}green_{i+1}] + colour_list[red_{i+1}{:}]$ $same\ colours\ as\ last\ iteration\ as\ before$

Since by inductive hypothesis we know that all([c == "b" for c in colour_list[0: $blue_i$]]) and $blue_{i+1} = blue_i$ hence all([c == "b" for c in colour_list[0: $blue_{i+1}$]])

Since by inductive hypothesis we know that all([c == "g" for c in colour_list[$blue_i$: $green_i$]]) and $green_{i+1} = green_i$ and $blue_{i+1} = blue_i$, hence all([c == "g" for c in colour_list[$blue_{i+1}$: $green_{i+1}$]])

Since by inductive hypothesis we know that all([c == "r" for c in colour_list[red_i:]]) and $red_{i+1} = red_i - 1$ and colour_list[red_i - 1] = "r", hence all([c == "r" for c in colour_list[red_{i+1}:]]).

So p(i+1) follows in this case.

- Case3: colour_list[red_i - 1] = "r" and colour_list[$green_i$] = "b".

According to the code $red_{i+1} = red_i - 1$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i + 1$.

And colour_list[$blue_i$], colour_list[$green_i$] = colour_list[$green_i$], colour_list[$blue_i$] that is the element on $blue_i$ and $green_i$ change to each other.

Since by inductive hypothesis we know that $0 \le blue_i \le green_i \le red_i \le len(colour\_list)$
And since we have the i+1 iteration, $green_i < red_i$
Hence $0 \le blue_i \le green_i < red_i \le len(colour\_list)$
Hence $red_{i+1} < red_i \le len(colour\_list)$, that is $red_{i+1} \le len(colour\_list)$
Also $0 \le blue_i < blue_{i+1} = blue_i + 1 \le green_i + 1 = green_{i+1}$, hence $0 \le blue_{i+1} \le green_{i+1}$
And since we have colour_list[red_i - 1] = "r" and colour_list[$green_i$] = "b" and they are not equal to each other, hence red_i $-1 > green_i$ that is red_i $-1 \ge green_i + 1$ that is $red_{i+1} \ge green_{i+1}$
Combining the above, we have $0 \le blue_{i+1} \le green_{i+1} \le red_{i+1} \le len(colour\_list)$
Since by inductive hypothesis we know that $\text{colour\_list}[0{:}green_i] + colour\_list[red_i{:}]\ same\ colours\ as\ before$ .Since we only change the element on $blue_i$ and $green_i$ to each other and they are all in $\text{colour\_list}[0{:}green_{i+1}]$ hence $\text{colour\_list}[0{:}green_{i+1}] + colour\_list[red_{i+1}{:}]\ same\ colours\ as\ the\ last\ iteration\ as\ before$
Since by inductive hypothesis we know that $\text{all([c == "b" for c in colour\_list[0:}blue_i\text{]])}$ and $blue_{i+1} = blue_i + 1\ and\ we've\ changed\ \text{colour\_list}[blue_i]\ \text{to}\ \text{colour\_list}[green_i]\ \text{which equals "b"}$ hence $\text{all([c == "b" for c in colour\_list[0:}blue_{i+1}\text{]])}$
Since by inductive hypothesis we know that $\text{all([c == "g" for c in colour\_list[}blue_i{:}green_i\text{]])}$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i + 1$ and colour_list[$green_i$] = "g" since we have changed it with $\text{colour\_list}[blue_i]$ which is "g" according to inductive hypothesis, hence $\text{all([c == "g" for c in colour\_list[}blue_{i+1}{:}green_{i+1}\text{]])}$
Since by inductive hypothesis we know that $\text{all([c == "r" for c in colour\_list[}red_i{:}\text{]])}$ and $red_{i+1} = red_i - 1$ and colour_list[red_i - 1] = "r", hence $\text{all([c == "r" for c in colour\_list[}red_{i+1}{:}\text{]])}$.
So p(i+1) follows in this case.

- Case4: colour_list[red_i - 1] = "b" or "g" and colour_list[$green_i$] = "g".

According to the code $red_{i+1} = red_i$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i$.
Since by inductive hypothesis we know that $0 \le blue_i \le green_i \le red_i \le len(colour\_list)$
And since we have the i+1 iteration, $green_i < red_i$
Hence $0 \le blue_i \le green_i < red_i \le len(colour\_list)$
Hence $red_{i+1} = red_i \le len(colour\_list)$, that is $red_{i+1} \le len(colour\_list)$
Also $0 \le blue_i = blue_{i+1} \le green_i < green_{i+1}$, hence $0 \le blue_{i+1} \le green_{i+1}$
And since $red_i > green_i$ that is $red_i \ge green_i + 1$ that is $red_{i+1} \ge green_{i+1}$
Combining the above, we have $0 \le blue_{i+1} \le green_{i+1} \le red_{i+1} \le len(colour\_list)$
Since by inductive hypothesis we know that $\text{colour\_list}[0{:}green_i] + colour\_list[red_i{:}]\ same\ colours\ as\ before$,Since we didn't change any element of the list in this case, $\text{colour\_list}[0{:}green_{i+1}] + colour\_list[red_{i+1}{:}]\ same\ colours\ as\ the\ last\ iteration\ as\ before$
Since by inductive hypothesis we know that $\text{all([c == "b" for c in colour\_list[0:}blue_i\text{]])}$ and $blue_{i+1} = blue_i$ hence $\text{all([c == "b" for c in colour\_list[0:}blue_{i+1}\text{]])}$
Since by inductive hypothesis we know that $\text{all([c == "g" for c in colour\_list[}blue_i{:}green_i\text{]])}$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i$ and colour_list[$green_i$] = "g", hence

all([c == "g" for c in colour_list[$blue_{i+1}$: $green_{i+1}$]])
Since by inductive hypothesis we know that all([c == "r" for c in colour_list[red_i:]]) and $red_{i+1} = red_i$, hence all([c == "r" for c in colour_list[red_{i+1}:]]).
So p(i+1) follows in this case.

- Case5: colour_list[red_i - 1] = "b" or "g" and colour_list[$green_i$] = "b".

According to the code $red_{i+1} = red_i$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i + 1$.
And colour_list[$blue_i$], colour_list[$green_i$] = colour_list[$green_i$], colour_list[$blue_i$] that is the element on $blue_i$ and $green_i$ change to each other.
Since by inductive hypothesis we know that $0 \leq blue_i \leq green_i \leq red_i \leq len(colour\_list)$
And since we have the i+1 iteration, $green_i < red_i$
Hence $0 \leq blue_i \leq green_i < red_i \leq len(colour\_list)$
Hence $red_{i+1} = red_i \leq len(colour\_list)$, that is $red_{i+1} \leq len(colour\_list)$
Also $0 \leq blue_i < blue_{i+1} = blue_i + 1 \leq green_i + 1 = green_{i+1}$, hence $0 \leq blue_{i+1} \leq green_{i+1}$
And since $red_i > green_i$ that is $red_i \geq green_i + 1$ that is $red_{i+1} \geq green_{i+1}$
Combining the above, we have $0 \leq blue_{i+1} \leq green_{i+1} \leq red_{i+1} \leq len(colour\_list)$
Since by inductive hypothesis we know that colour_list[0: $green_i$] + $colour\_list[red_i:]\ same\ colours\ as\ before$ .Since we only change the element on $blue_i$ and $green_i$ to each other and they are all in colour_list[0: $green_{i+1}$] hence colour_list[0: $green_{i+1}$] + $colour\_list[red_{i+1}:]\ same\ colours\ as\ the\ last\ iteration\ as\ before$
Since by inductive hypothesis we know that all([c == "b" for c in colour_list[0: $blue_i$]]) and $blue_{i+1} = blue_i + 1$ $and\ we've\ changed$ colour_list[$blue_i$] to colour_list[$green_i$] which equals "b" hence all([c == "b" for c in colour_list[0: $blue_{i+1}$]])
Since by inductive hypothesis we know that all([c == "g" for c in colour_list[$blue_i$: $green_i$]]) and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i + 1$ and colour_list[$green_i$] = "g" since we have changed it with colour_list[$blue_i$] which is "g" according to inductive hypothesis, hence all([c == "g" for c in colour_list[$blue_{i+1}$: $green_{i+1}$]])
Since by inductive hypothesis we know that all([c == "r" for c in colour_list[red_i:]]) and $red_{i+1} = red_i$, hence all([c == "r" for c in colour_list[red_{i+1}:]]).
So p(i+1) follows in this case.

- Case6: colour_list[red_i - 1] = "g" and colour_list[$green_i$] = "r".

According to the code $red_{i+1} = red_i - 1$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i$.
And colour_list[red_i - 1], colour_list[$green_i$] = colour_list[$green_i$], colour_list[red_i - 1] that is the element on red_i - 1 and $green_i$ change to each other.
Since by inductive hypothesis we know that $0 \leq blue_i \leq green_i \leq red_i \leq len(colour\_list)$
And since we have the i+1 iteration, $green_i < red_i$
Hence $0 \leq blue_i \leq green_i < red_i \leq len(colour\_list)$
Hence $red_{i+1} < red_i \leq len(colour\_list)$, that is $red_{i+1} \leq len(colour\_list)$
Also $0 \leq blue_i = blue_{i+1} \leq green_i + 1 = green_{i+1}$, hence $0 \leq blue_{i+1} \leq green_{i+1}$
And since we have colour_list[red_i - 1] = "g" and colour_list[$green_i$] = "r" and they are not equal to each other, hence $red_i - 1 > green_i$ that is $red_i - 1 \geq green_i + 1$ that is $red_{i+1} \geq green_{i+1}$

Combining the above, we have $0 \leq blue_{i+1} \leq green_{i+1} \leq red_{i+1} \leq len(colour\_list)$
Since by inductive hypothesis we know that $\text{colour\_list}[0{:}green_i] + colour\_list[red_i{:}]\ same\ colours\ as\ before$ .Since we only change the element on $red_i - 1$ and $green_i$ to each other and colour_list[red_i - 1] in $colour\_list[red_{i+1}{:}]$ before and now in $\text{colour\_list}[0{:}green_{i+1}]$, And colour_list[$green_i$] in $\text{colour\_list}[0{:}green_{i+1}]\ before\ and\ now\ in\ colour\_list[red_{i+1}{:}]$ hence $\text{colour\_list}[0{:}green_{i+1}] +$
$colour\_list[red_{i+1}{:}]\ same\ colours\ as\ the\ last\ iteration\ as\ before$.
Since by inductive hypothesis we know that $\text{all}([\text{c} == \text{"b" for c in colour\_list}[0{:}blue_i]])$ and $blue_{i+1} = blue_i$ hence $\text{all}([\text{c} == \text{"b" for c in colour\_list}[0{:}blue_{i+1}]])$
Since by inductive hypothesis we know that $\text{all}([\text{c} == \text{"g" for c in colour\_list}[blue_i{:}green_i]])$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i$ and colour_list[$green_i$] = "g" since we have changed it with $\text{colour\_list}[red_i - 1]$ which is "g", hence $\text{all}([\text{c} == \text{"g" for c in colour\_list}[blue_{i+1}{:}green_{i+1}]])$
Since by inductive hypothesis we know that $\text{all}([\text{c} == \text{"r" for c in colour\_list}[red_i{:}]])$ and $red_{i+1} = red_i - 1$ and colour_list[red_i - 1] = "r" since we have changed it with $\text{colour\_list}[green_i]$ which is "r", hence $\text{all}([\text{c} == \text{"r" for c in colour\_list}[red_{i+1}{:}]])$.
So p(i+1) follows in this case.

- Case7: colour_list[red_i - 1] = "b" and colour_list[$green_i$] = "r".

According to the code $red_{i+1} = red_i - 1$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i + 1$.
And colour_list[red_i - 1], colour_list[$green_i$] = colour_list[$green_i$], colour_list[red_i - 1] that is the element on red_i - 1 and $green_i$ change to each other.
And then colour_list[$blue_i$], colour_list[$green_i$] = colour_list[$green_i$], colour_list[$blue_i$] that is the element on $blue_i$ and $green_i$ change to each other.
That is colour_list[red_i - 1] is the colour_list[$green_i$] of the last list.
colour_list[$green_i$] is the colour_list[$blue_i$] of the last list.
colour_list[$blue_i$] is the colour_list[red_i - 1] of the last list.
Since by inductive hypothesis we know that $0 \leq blue_i \leq green_i \leq red_i \leq len(colour\_list)$
And since we have the i+1 iteration, $green_i < red_i$
Hence $0 \leq blue_i \leq green_i < red_i \leq len(colour\_list)$
Hence $red_{i+1} < red_i \leq len(colour\_list)$, that is $red_{i+1} \leq len(colour\_list)$
Also $0 \leq blue_i < blue_{i+1} = blue_i + 1 \leq green_i + 1 = green_{i+1}$, hence $0 \leq blue_{i+1} \leq green_{i+1}$
And since we have colour_list[red_i - 1] = "b" and colour_list[$green_i$] = "r" and they are not equal to each other, hence $red_i - 1 > green_i$ that is $red_i - 1 \geq green_i + 1$ that is $red_{i+1} \geq green_{i+1}$
Combining the above, we have $0 \leq blue_{i+1} \leq green_{i+1} \leq red_{i+1} \leq len(colour\_list)$
Since by inductive hypothesis we know that $\text{colour\_list}[0{:}green_i] + colour\_list[red_i{:}]\ same\ colours\ as\ before$ .Since colour_list[red_i - 1] is the colour_list[$green_i$] of the last list. colour_list[$green_i$] is the colour_list[$blue_i$] of the last list. colour_list[$blue_i$] is the colour_list[red_i - 1] of the last list. And they are all in $\text{colour\_list}[0{:}green_{i+1}] + colour\_list[red_{i+1}{:}]$,
hence $\text{colour\_list}[0{:}green_{i+1}] +$

$colour\_list[red_{i+1}:]$ $same\ colours\ as\ the\ last\ iteration\ as\ before$.
Since by inductive hypothesis we know that $\text{all([c == "b" for c in colour\_list[0:}blue_i\text{]])}$ and $blue_{i+1} = blue_i + 1$ and colour_list[$blue_i$] is the colour_list[red_i − 1] of the last list which is "b" hence $\text{all([c == "b" for c in colour\_list[0:}blue_{i+1}\text{]])}$
Since by inductive hypothesis we know that $\text{all([c == "g" for c in colour\_list[}blue_i: green_i\text{]])}$ and $green_{i+1} = green_i + 1$ and $blue_{i+1} = blue_i + 1$ and colour_list[$green_i$] = "g" colour_list[$green_i$] is the colour_list[$blue_i$] of the last list, and by inducitve hypothesis, the colour_list[$blue_i$] of the last list is "g", hence $\text{all([c == "g" for c in colour\_list[}blue_{i+1}: green_{i+1}\text{]])}$
Since by inductive hypothesis we know that $\text{all([c == "r" for c in colour\_list[}red_i:\text{]])}$ and $red_{i+1} = red_i - 1$ and colour_list[red_i - 1] = "r" since we have changed it with colour_list[$green_i$] which is "r", hence $\text{all([c == "r" for c in colour\_list[}red_{i+1}:\text{]])}$.
So p(i+1) follows in this case.
Hence p(i+1) follows.
Hence $\forall \text{i} \in \mathbb{N}, \text{p(i)}$.
If the loop terminates after, say, iteration f, then the following must be true:

- $0 \le blue_f \le green_f \le red_f \le len(colour\_list)$ and colour_list[0: $green_f$] + $colour\_list[red_f:]$ $same\ colours\ as\ before$ and $\text{all([c == "b" for c in colour\_list[0:}blue_f\text{]])}$ and $\text{all([c == "g" for c in colour\_list[}blue_f: green_f\text{]])}$ and $\text{all([c == "r" for c in colour\_list[}red_f:\text{]])}$ #by p(f)
- $green_f \ge red_f$ #by loop condition.

Hence $0 \le blue_f \le green_f = red_f \le len(colour\_list)$ and $\text{all([c == "b" for c in colour\_list[0:}blue_f\text{]])}$ and $\text{all([c == "g" for c in colour\_list[}blue_f: green_f\text{]])}$ and $\text{all([c == "r" for c in colour\_list[}red_f:\text{]])}$ and colour_list[0: $green_f$] + $colour\_list[red_f:]$ $same\ colours\ as\ before$.
Hence we can conclude the postcondition that is colour_list has same strings as before, ordered "b" < "g" < "r".

In the end, we will prove termination:
Try the sequence <$red_i - green_i$>, by initialization and the code , it is obviously that $red_i \in \mathbb{N}$, $green_i \in \mathbb{N}$ .By the loop invariant p(i), $green_i \le red_i$. Hence $red_i - green_i$ is an integer, and $green_i \le red_i \Longrightarrow red_i - green_i \ge 0$, so each element of the sequence is $\in \mathbb{N}$.
It remains to show that the sequence is strictly decreasing. Suppose there is an (i+1)th iteration of the loop. Then $red_{i+1} - green_{i+1} \le red_i - green_i - 1 < red_i - green_i$ (since from the above case1,3,4,5,6,7 we know that $green_{i+1}$ will always equal to $green_i + 1$ while red_{i+1} will either remain the same as red_i or equal to $red_i - 1$. for case2, however $red_{i+1} = red_i - 1$ while $green_{i+1} =$ $green_i$ but it concludes the same that $red_{i+1} - green_{i+1} \le red_i - green_i - 1$) . So the sequence is strictly decreasing.
Thus the loop terminates.
Hence my function is right.■