Csc236













2.

(a)

Answer:

The explanation is the following:

Since , hence p(0)=1,

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

Since , hence p(3)=p(4)=p(5)=1,

Since , hence p(6)=p(7)=1,

Since , hence p(8)=p(9) =p(10)=p(11)=2,

Since , 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)

ways to create n with the use of 3-cent stamp.

ways to create n with the use of 4-cent stamp.

ways to create n with the use of 5-cent stamp.

ways to create n with the use of 3 and4-cent stamp.

ways to create n with the use of 4 and5-cent stamp.

ways to create n with the use of 3and5-cent stamp.

ways to create n with the use of 3,4and5-cent stamp.

Hence when n>12,

Hence

(b)

WTS： p(n) is monotonic nondecreasing

Proof:

We are going to prove that p(n) is monotonic nondecreasing. That is to prove that p(n)p(n-1). That is p(n)-p(n-1).

Define q(n) p(n)-p(n-1). So we are going to prove that q(n)

I am going to prove this by complete induction.

Inductive step:

.

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

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

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)

Hence, we only need to show that

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

hence , and I can prove that with the following picture:

P(n-6)

P(n-7)

P(n-8)

P(n-10)

P(n-12)

P(n-11)

P(n-15)

We can see that is actually the blue space, with be the white space, and the combination of them is p(n-3).

Since the blue space must have non-negative ways hence must be non-negative, that is That is

Hence

So q(n) follows in this case.

Hence we have proved that p(n) is monotonic nondecreasing.









Define

Then





4.

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

Proof:

I will use complete induction to prove

Inductive step:

.

* 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 is 1, hence the result is right.
2. i>0=j, then according to the code, it will return 0, since 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 is 1, hence the result is right.
2. i>j, then according to the code, it will return 0, since is 0, hence the result is right.
3. , then according to the code, it will return count\_subsequence(s1,s2,i,j-1), since j>0, , hence according to the inducive hypothesis, since , hence , hence .

Since 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 , in other word, we only need to consider about , hence the is equal to the , hence the result is right.

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, , hence according to the inducive hypothesis, since , hence , hence . Since 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 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 , hence we only need to consider the numbers of . Hence in this situation the number is equal to the numbers of . In the other situation, I will not use as a term of the subsequence, hence we only need to consider about the numbers of , hence in this situation the number is equal to the numbers of . So we combinate the 2 situation together and the total number is

hence the result is right.

So c(j) follows in this case

Hence we have proved that, 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”}.

.

Define p(i): after the ith iteration of the loop (if it occurs), and and

and and

I am going to prove

Base case: (by initialization),

Since , hence

and it has the same colours as before since it is the 0th iteration.

must always be true.

must always be true.

must always be true.

So p(0) follows.

Inductive step:

Let 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[ - 1] = "r" and colour\_list[] = "g".

According to the code and and .

Since by inductive hypothesis we know that

And since we have the i+1 iteration,

Hence

Hence , that is

Also , hence

And since we have colour\_list[ - 1] = "r" and colour\_list[] = "g" and they are not equal to each other, hence – 1> that is – 1 that is

Combining the above, we have

Since by inductive hypothesis we know that .Since we didn’t change any element of the list in this case,

Since by inductive hypothesis we know that and hence

Since by inductive hypothesis we know that and and and colour\_list[] = "g", hence

Since by inductive hypothesis we know that and and colour\_list[ - 1] = "r", hence .

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

* Case2: colour\_list[ - 1] = "r" and colour\_list[] = "r".

According to the code and and .

Since by inductive hypothesis we know that

And since we have the i+1 iteration,

Hence

Hence , that is

Also , hence

And since > that is that is

Combining the above, we have

Since by inductive hypothesis we know that .Since we didn’t change any element of the list in this case,

Since by inductive hypothesis we know that and hence

Since by inductive hypothesis we know that and and, hence

Since by inductive hypothesis we know that and and colour\_list[ - 1] = "r", hence .

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

* Case3: colour\_list[ - 1] = "r" and colour\_list[] = "b".

According to the code and and .

And colour\_list[], colour\_list[] = colour\_list[], colour\_list[] that is the element on and change to each other.

Since by inductive hypothesis we know that

And since we have the i+1 iteration,

Hence

Hence , that is

Also , hence

And since we have colour\_list[ - 1] = "r" and colour\_list[] = "b" and they are not equal to each other, hence – 1> that is – 1 that is

Combining the above, we have

Since by inductive hypothesis we know that .Since we only change the element on and to each other and they are all inhence

Since by inductive hypothesis we know that and hence

Since by inductive hypothesis we know that and and and colour\_list[] = "g" since we have changed it with which is “g” according to inductive hypothesis, hence

Since by inductive hypothesis we know that and and colour\_list[ - 1] = "r", hence .

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

* Case4: colour\_list[ - 1] = "b" or “g” and colour\_list[] = "g".

According to the code and and .

Since by inductive hypothesis we know that

And since we have the i+1 iteration,

Hence

Hence , that is

Also , hence

And since > that is that is

Combining the above, we have

Since by inductive hypothesis we know that ,Since we didn’t change any element of the list in this case,

Since by inductive hypothesis we know that and hence

Since by inductive hypothesis we know that and and and colour\_list[] = "g", hence

Since by inductive hypothesis we know that and , hence .

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

* Case5: colour\_list[ - 1] = "b" or “g” and colour\_list[] = "b".

According to the code and and .

And colour\_list[], colour\_list[] = colour\_list[], colour\_list[] that is the element on and change to each other.

Since by inductive hypothesis we know that

And since we have the i+1 iteration,

Hence

Hence , that is

Also , hence

And since > that is that is

Combining the above, we have

Since by inductive hypothesis we know that .Since we only change the element on and to each other and they are all inhence

Since by inductive hypothesis we know that and hence

Since by inductive hypothesis we know that and and and colour\_list[] = "g" since we have changed it with which is “g” according to inductive hypothesis, hence

Since by inductive hypothesis we know that and , hence .

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

* Case6: colour\_list[ - 1] = “g” and colour\_list[] = "r".

According to the code and and .

And colour\_list[ - 1], colour\_list[] = colour\_list[], colour\_list[ - 1] that is the element on - 1 and change to each other.

Since by inductive hypothesis we know that

And since we have the i+1 iteration,

Hence

Hence , that is

Also , hence

And since we have colour\_list[ - 1] = "g" and colour\_list[] = "r" and they are not equal to each other, hence – 1> that is – 1 that is

Combining the above, we have

Since by inductive hypothesis we know that .Since we only change the element on – 1 and to each other and colour\_list[ - 1] in before and now in , And colour\_list[] inhence.

Since by inductive hypothesis we know that and hence

Since by inductive hypothesis we know that and and and colour\_list[] = "g" since we have changed it with which is “g”, hence

Since by inductive hypothesis we know that and and colour\_list[ - 1] = "r" since we have changed it with which is “r”, hence .

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

* Case7: colour\_list[ - 1] = “b” and colour\_list[] = "r".

According to the code and and .

And colour\_list[ - 1], colour\_list[] = colour\_list[], colour\_list[ - 1] that is the element on - 1 and change to each other.

And then colour\_list[], colour\_list[] = colour\_list[], colour\_list[] that is the element on and change to each other.

That is colour\_list[ - 1] is the colour\_list[] of the last list.

colour\_list[] is the colour\_list[] of the last list.

colour\_list[] is the colour\_list[ - 1] of the last list.

Since by inductive hypothesis we know that

And since we have the i+1 iteration,

Hence

Hence , that is

Also , hence

And since we have colour\_list[ - 1] = "b" and colour\_list[] = "r" and they are not equal to each other, hence – 1> that is – 1 that is

Combining the above, we have

Since by inductive hypothesis we know that .Since colour\_list[ - 1] is the colour\_list[] of the last list. colour\_list[] is the colour\_list[] of the last list. colour\_list[] is the colour\_list[ - 1] of the last list. And they are all in ,

hence.

Since by inductive hypothesis we know that and hence

Since by inductive hypothesis we know that and and and colour\_list[] = "g" colour\_list[] is the colour\_list[] of the last list, and by inducitve hypothesis, the colour\_list[] of the last list is “g”, hence

Since by inductive hypothesis we know that and and colour\_list[ - 1] = "r" since we have changed it with which is “r”, hence .

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

Hence p(i+1) follows.

Hence .

If the loop terminates after, say, iteration f, then the following must be true:

* and and

and and #by p(f)

* #by loop condition.

Hence and and and and .

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 <>, by initialization and the code , it is obviously that .By the loop invariant p(i), . Hence is an integer, and , so each element of the sequence is .

It remains to show that the sequence is strictly decreasing. Suppose there is an (i+1)th iteration of the loop. Then (since from the above case1,3,4,5,6,7 we know that will always equal to while will either remain the same as or equal to . for case2, however while but it concludes the same that ) . So the sequence is strictly decreasing.

Thus the loop terminates.

Hence my function is right.