

Append:

[Append.txt](#)

[Store and Environment](#)

```
// From the information of the store, when we pass L3: (7|(8|(9|(10|nil)))) into Reverse,  
// it is doing:  
// 1. {Append {Reverse [8,9,10]} [7]} -----> {Append [10,9,8] [7]} -----> [10,9,8,7] (final result)  
// 2. {Reverse [8,9,10]} ----> {Append {Reverse [9,10]} [8]} -----> {Append [10,9] [8]}  
// 3. {Reverse [9,10]} ----> {Append {Reverse [10]} [9]} -----> {Append [10] [9]} -----> [10,9]  
// 4. {Reverse [10]} ----> {Append {Reverse []} [10]} -----> [10]
```

Append_diff:

[Append_diff.txt](#)

[Store and Environment](#)

```
// From the information of the store, when we pass [4,3,2,1] into Reverse  
// {Reverse [4,3,2,1]} -----> {ReverseD [4,3,2,1] Out []} -----> Return [1,2,3,4]  
// {ReverseD [4,3,2,1] Out []} -----> {ReverseD [3,2,1] Out [4]}  
// {ReverseD [3,2,1] Out [4]} -----> {ReverseD [2,1] Out [3,4]}  
// {ReverseD [2,1] Out [3,4]} -----> {ReverseD [1] Out [2,3,4]}  
// {ReverseD [1] Out [2,3,4]} -----> {ReverseD [] Out [1,2,3,4]}  
// {ReverseD [] Out [1,2,3,4]} -----> for the [] case, ReverseD bound [1,2,3,4] to Out, Out = [1,2,3,4]  
// Return [1,2,3,4]
```

Reverse with size of 6:

For Reverse in append.txt:

Reverse ([1,2,3,4,5,6])

Append (reverse ([2,3,4,5,6]) [1])

Reverse ([2,3,4,5,6]) = Append (reverse ([3,4,5,6]) [2])

Reverse ([3,4,5,6]) = Append (reverse ([4,5,6]) [3])

Reverse ([4,5,6]) = Append (reverse ([5,6]) [4])

Reverse ([5,6]) = Append (reverse ([6]) [5])

Reverse ([6]) = Append (Reverse [] [6])

Reverse [] = []

So, For Reverse in append.txt, the number of cons ']' operations should be:

$$6+5+4+3+2+1=21$$

For Reverse in append_diff.txt:

Reverse ([1,2,3,4,5,6] [])

Reverse ([2,3,4,5,6] [1])

Reverse ([3,4,5,6] [2,1])

Reverse ([4,5,6] [3,2,1])

Reverse ([5,6] [4,3,2,1])

Reverse ([6] [5,4,3,2,1])

Reverse ([] [6,5,4,3,2,1]) = [6,5,4,3,2,1]

So, For Reverse in append_diff.txt, the number of cons ']' operations should be:

$$1+1+1+1+1+1 = 7$$