End to Beginning Algorithm

```
    End_to_Beginning (A[], n)
1 — [ H = new int[n]; 1

      //Populate H with Ø values
n — [ for(int i=0; i<n; i++)
          H[i]=0; 1
```

$$\sum_{i=0}^{n-1} 1 = (n-1-0+1)*1 = \boxed{n}$$

```
         //loop to calculate values of H
n²-n-4 — [ for(int i=n-2; i>=0; i--)
               for(int K=i+1; K<n; K++) {
2=max(3,0)         [ if (A[i] ≤ A[K] & H[i] <= H[K])
                         H[i] = H[K]+1; 1+1=2
               }
           3
```

$$\sum_{k=i+1}^{n-1} 2 = (n-1-i-\cancel{1}\cancel{+1})*2 \rightarrow 2n-2-2i$$

$$\sum_{i=0}^{n-2}(2n-2i-2) = \sum_{i=0}^{n-2} 2n + \sum_{i=0}^{n-2} 2i - \sum_{i=0}^{n-2} 2 = 2n(n-1) - 2(n-1) - 2 * \frac{(n-2)(n-2+1)}{2}$$

```
         //calculate max of longest subsequence +1
1 — [ max = H[0]
2n-2 = [ for(int i=1; i<n; i++)
              if(max < H[i]) 1      max(1,0)+1 = 2
                  max=H[i];
       max++;
```

$$\sum_{i=1}^{n-1} 2 = 2(n-1-\cancel{1}+\cancel{1}) = 2n-2$$

```
         //allocate space for R
1 - [ R = new float[max];

              //Populate R
2 = 1+1 = [ int val= max-1;
4n = [ for (int i=0, K=0; i<n; i++){
            if(H[i] == val)   1+max(3,0)=4
                R[K] = A[i]; 1
                val--; 1
                K++; 1
       }
```

$$\sum_{i=0}^{n-1} 4 = 4(n-1-0+1) = 4n$$

End to beginning Algorithm

$\boxed{\cancel{1}}+\underline{n}+\underline{n^2}-\underline{n}\boxed{\cancel{-4}}+\boxed{1}+\underline{2n}\boxed{\cancel{-2}}+\boxed{\cancel{1}}+\boxed{2}+\underline{4n}$

n^2+6n-1

$n^2+6n-1 \in O(n^2)$

Proof:

$$\lim_{n\to\infty} \frac{n^2+6n-1}{n^2} \Rightarrow \frac{1+\cancelto{0}{\frac{6}{n}}-\cancelto{0}{\frac{1}{n^2}}}{1} = 1$$

$$\lim_{n\to\infty} \frac{n^2+6n-1}{n^2} = 1 > 0 \text{ and constant}$$

therefore

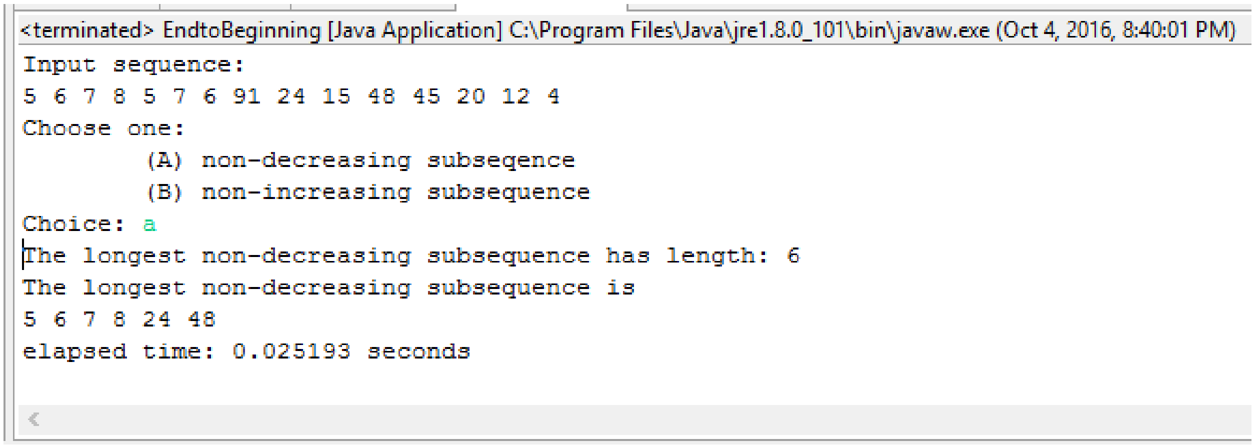
$n^2+6n-1 \in O(n^2)$

and

End to beginning Algorithm has

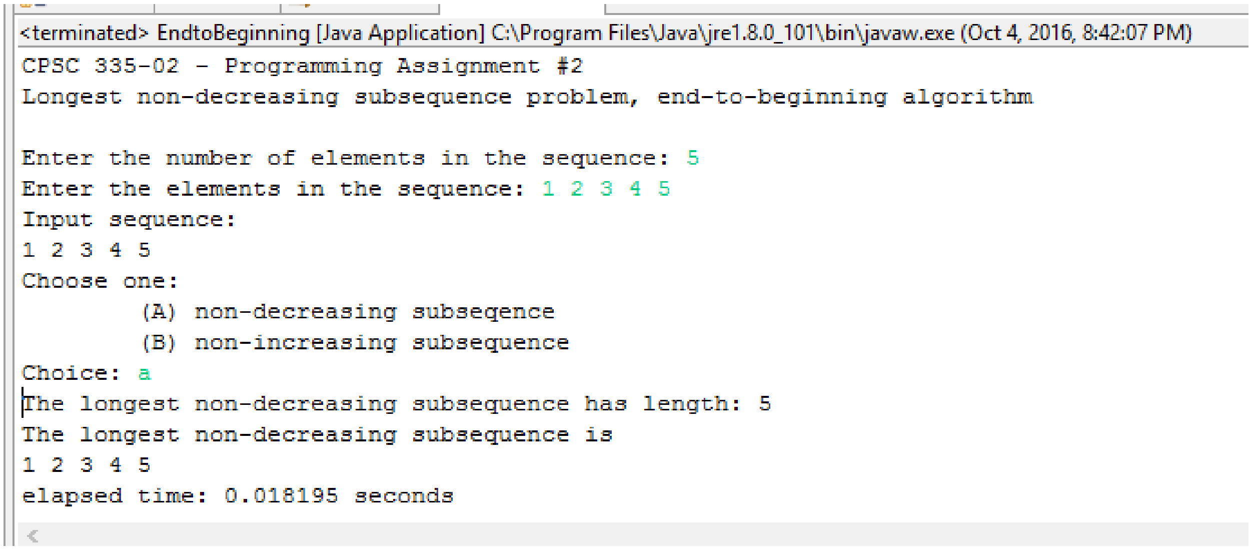
$O(n^2)$

End to beginning algorithm



<terminated> EndtoBeginning [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Oct 4, 2016, 8:40:01 PM)

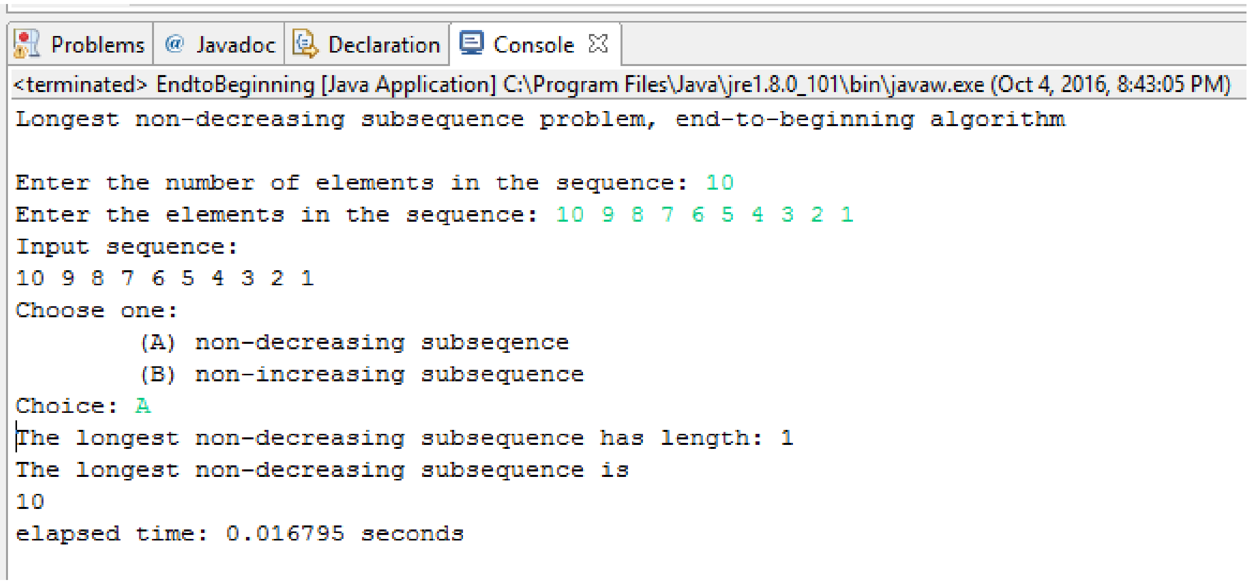
```
Input sequence:
5 6 7 8 5 7 6 91 24 15 48 45 20 12 4
Choose one:
        (A) non-decreasing subseqence
        (B) non-increasing subsequence
Choice: a
The longest non-decreasing subsequence has length: 6
The longest non-decreasing subsequence is
5 6 7 8 24 48
elapsed time: 0.025193 seconds
```



<terminated> EndtoBeginning [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Oct 4, 2016, 8:42:07 PM)

```
CPSC 335-02 - Programming Assignment #2
Longest non-decreasing subsequence problem, end-to-beginning algorithm

Enter the number of elements in the sequence: 5
Enter the elements in the sequence: 1 2 3 4 5
Input sequence:
1 2 3 4 5
Choose one:
        (A) non-decreasing subseqence
        (B) non-increasing subsequence
Choice: a
The longest non-decreasing subsequence has length: 5
The longest non-decreasing subsequence is
1 2 3 4 5
elapsed time: 0.018195 seconds
```



<terminated> EndtoBeginning [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Oct 4, 2016, 8:43:05 PM)

```
Longest non-decreasing subsequence problem, end-to-beginning algorithm

Enter the number of elements in the sequence: 10
Enter the elements in the sequence: 10 9 8 7 6 5 4 3 2 1
Input sequence:
10 9 8 7 6 5 4 3 2 1
Choose one:
        (A) non-decreasing subseqence
        (B) non-increasing subsequence
Choice: A
The longest non-decreasing subsequence has length: 1
The longest non-decreasing subsequence is
10
elapsed time: 0.016795 seconds
```