Longest Increasing Subsequence

Longest Increasing Subsequence

- Problem (longest increasing subsequence): You are given a sequence of integers $A[1],\ldots,A[n]$ and you are asked to find the ***longest increasing subsequence of integers***.
- Example: The longest increasing subsequence of the sequence (7,2,8,6,3,6,9,7) is (2,3,6,7).
- Let $L(i)$ denote the length of the longest increasing subsequence that ends with the number $A[i]$.
- What is $L(1)$?
  - $L1=1$

- What is the value of $L(i)$ in terms of $L(1),\ldots,L(i-1)$?
- $L(i)={}_{j<i,A[j]\leq A[i]}\max(1+L(j))$.

Example

- Let n=9 and $(A[1],\ldots,A[9])$ = (7,2,8,6,3,1,9,7,10).
- $L(1)$ = 1
- $L(2)$ = 1
- $L(3)$ = 2
- $L(4)$ = 2
- $L(5)$ = 2
- $L(6)$ = 1
- $L(7)$ = max (2,2,3,3,3,2) = 3
- $L(8)$ = max (2,2,3,3,2) = 3
- $L(9)$ = max (2,2,3,3,3,2,4,4) = 4
- What is the length of the longest increasing subsequence?
  - $\max_{1\leq j\leq n} L(j)$

Algorithm

```
Length-LIS(A)
for i = 1 to n
   max←1
   for j = 1 to (i−1)
        if (A[j] ≤ A[i])
               if (max< L[j]+1)
                       max← L[j]+1
   L[i] =max
 Return the maximum of L[i]'s
```

- What is the running time for the above algorithm?
- $T(n) = O(n^2)$

- But the problem was to find the longest increasing subsequence and not the length
- **For each number, we just note down the index of the number preceding this number in a longest increasing subsequence.**

Algorithm

```
Length-LIS(A)
for i = 1 to n
    max←1
    P[i] ← 1
    for j = 1 to (i−1)
        if (A[j] ≤ A[i])
            if (max< L[j]+1)
                max← L[j]+1
                P[i] = j
    L[i] =max
```

P stores the longest increasing subsequence output this

	1	2	3	4	5	6	7	8	9
A	7	2	8	6	3	1	9	7	10
L	1	1	2	2	2	1	3	3	4
P	1	2	1	2	2	6	3	4	7

```
Length-LIS(A)
for i = 1 to n
        max←1
        P[i] ← 1
        for j = 1 to (i−1)
                if (A[j] ≤ A[i])
                        if (max< L[j]+1)
                                max← L[j]+1
                                P[i] = j
        L[i] =max
```

P stores the longest increasing subsequence output this

Dynamic Programming: Examples

	1	2	3	4	5	6	7	8	9
A	7	2	8	6	3	1	9	7	10
L	1	1	2	2	2	1	3	3	4
P	1	2	1	2	2	6	3	4	7

So one of the longest increasing subsequence is (7, 8, 9, 10).