

Algorithm Lab

Week 5: Longest Increasing Subsequence

Problem

<http://oj.csie.ndhu.edu.tw/problem/ALG04B> (<http://oj.csie.ndhu.edu.tw/problem/ALG04B>).

Solution

```

1  #include<iostream>
2  #include<cstdio>
3  #include<algorithm>
4  #include<cstring>
5  using namespace std;
6  int n, a[10100], len1[10100], len2[10100], b[10100];
7  void L(int len[],int a[])
8  {
9      int dp[10100];
10     int t=0;
11     dp[t]=-1;
12     for(int i=1; i<=n; i++){
13         if(a[i]>dp[t]){//If a[i]> the top element of the stack, push the s
14             dp[++t]=a[i];
15             len[i]=t;
16         }
17         else{//If a[i] is not greater than the top element of the stack, l
18             int l=1,r=t;
19             while(l<=r){
20                 int m=(l+r)/2;
21                 if(a[i]>dp[m])
22                     l=m+1;
23                 else
24                     r=m-1;
25             }
26             //replace a[i]
27             dp[l]=a[i];
28             len[i]=l;
29         }
30     }
31 }
32 int main(){
33     int i, j, s, mmax, ans;
34     while(~scanf("%d",&n)){
35         for(i=1; i<=n; i++){
36             scanf("%d",&a[i]);
37             b[n-i+1] = a[i];
38             len1[i] = 0;
39             len2[i] = 0;
40         }
41         L(len1,a);
42         L(len2,b);
43         mmax = -1;
44         ans = 0;
45         for(i=1; i<=n; i++){
46             ans = min(len1[i],len2[n-i+1])*2-1;
47             mmax = max(mmax, ans);
48         }
49         printf("%d\n",mmax);
50     }
51     return 0;
52 }

```

Time Complexity

Inserting the sequence into array = $O(N)$

Finding longest increasing subsequence = $O(N \log(N))$

Finding longest decreasing subsequence = $O(N \log(N))$

Finding longest wavio sequence = $O(N)$

Total time complexity = $O(N \log(N))$

Space Complexity

The algorithm uses 4 auxiliary arrays of size N ,
so the space complexity = $O(N)$