

Contents

1	DP	1
1.1	LCS	1
1.2	LIS $O(n^2)$	1
1.3	LIS $O(n \log n)$	1
1.4	LIS $O(n \log n)$	1

1 DP

1.1 LCS

```

1 #include <bits/stdc++.h>
2 #define IOS
3   ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
4 using namespace std;
5 string s1, s2;
6 int dp[505][505];
7 int main(){
8   IOS
9   cin >> s1 >> s2;
10  memset(dp, 0, sizeof(dp));
11  int l1 = s1.size(), l2 = s2.size();
12  for(int i = 1; i <= l1; i++){
13    for(int j = 1; j <= l2; j++){
14      if(s1[i - 1] == s2[j - 1]) dp[i][j] =
15        dp[i - 1][j - 1] + 1;
16      else dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
17    }
18  }
19  cout << dp[l1][l2] << '\n';
20  return 0;
21 }
```

1.2 LIS $O(n^2)$

```

1 #include <bits/stdc++.h>
2 #define IOS
3   ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
4 using namespace std;
5 typedef long long ll;
6 int main(){
7   IOS
8   int arr[100];
9   int n;
10  cin >> n;
11  for(int i = 0; i < n; i++) cin >> arr[i];
12  int dp[100];
13  for(int i = 0; i < n; i++) dp[i] = 1;
14  for(int i = 0; i < n; i++){
15    for(int j = 0; j < i; j++){
16      if(arr[i] > arr[j])
17        dp[i] = max(dp[j] + 1, dp[i]);
18    }
19  }
20  int ans = 1;
21  for(int i = 0; i < n; i++) ans = max(ans, dp[i]);
22  cout << ans << '\n';
23  return 0;
24 }
```

1.3 LIS $O(n \log n)$

```

1 class Solution {
2 public:
3   int lengthOfLIS(vector<int>& nums) {
4     vector<int> v;
```

```

5     int n = nums.size();
6     for(int i = 0; i < n; i++){
7       int p = lower_bound(v.begin(), v.end(),
8         nums[i]) - v.begin();
9       if(p == v.size()) v.push_back(nums[i]);
10      else v[p] = nums[i];
11    }
12    return v.size();
13  };
14 }
```

1.4 LIS $O(n \log n)$

```

1 for(int i=0;i<num.size();i++){
2   if(lis.empty()||lis.back()<num[i]){
3     lis.push_back(num[i]);
4     dp[i]=lis.size();
5   }
6   else{
7     auto iter=lower_bound(all(lis),num[i]);
8     dp[i]=iter-lis.begin()+1;
9     *iter=num[i];
10  }
11 }
12 int length=lis.size();
13 for(int i=num.size()-1;i>=0;i--){
14   if(dp[i]==length){
15     ans.push_back(num[i]);
16     length--;
17   }
18 }
```