Contents

1 DP

1.1 LCS

```cpp
#include <bits/stdc++.h>
#define IOS
    ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
using namespace std;
string s1, s2;
int dp[505][505];
int main(){
    IOS
    cin >> s1 >> s2;
    memset(dp, 0, sizeof(dp));
    int l1 = s1.size(), l2 = s2.size();
    for(int i = 1;i <= l1;i++){
        for(int j = 1;j <= l2;j++){
            if(s1[i - 1] == s2[j - 1]) dp[i][j] =
                dp[i - 1][j - 1] + 1;
            else dp[i][j] = max(dp[i - 1][j], dp[i][j
                - 1]);
        }
    }
    cout << dp[l1][l2] << '\n';

    return 0;
}
```

1.2 LIS $O(n^2)$

```cpp
#include <bits/stdc++.h>
#define IOS
    ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
using namespace std;
typedef long long ll;
int main(){
    IOS
    int arr[100];
    int n;
    cin >> n;
    for(int i = 0;i < n;i++) cin >> arr[i];
    int dp[100];
    for(int i = 0;i < n;i++) dp[i] = 1;
    for(int i = 0;i < n;i++){
        for(int j = 0;j < i;j++){
            if(arr[i] > arr[j])
                dp[i] = max(dp[j] + 1, dp[i]);
        }
    }
    int ans = 1;
    for(int i = 0;i < n;i++) ans = max(ans, dp[i]);
    cout << ans << '\n';

    return 0;
}
```

1.3 LIS $O(n \log n)$

```cpp
class Solution {
public:
    int lengthOfLIS(vector<int>& nums) {
        vector<int> v;
        int n = nums.size();
        for(int i = 0;i < n;i++){
            int p = lower_bound(v.begin(), v.end(),
                nums[i]) - v.begin();
            if(p == v.size()) v.push_back(nums[i]);
            else v[p] = nums[i];
        }
        return v.size();
    }
};
```