ALGOMANIACS

Dynamic Programming

Shobhit Gupta

TOPICS TO BE COVERED

- How to approach
- Iterative and recursive
- Time complexity
- Questions

HOW TO APPROACH

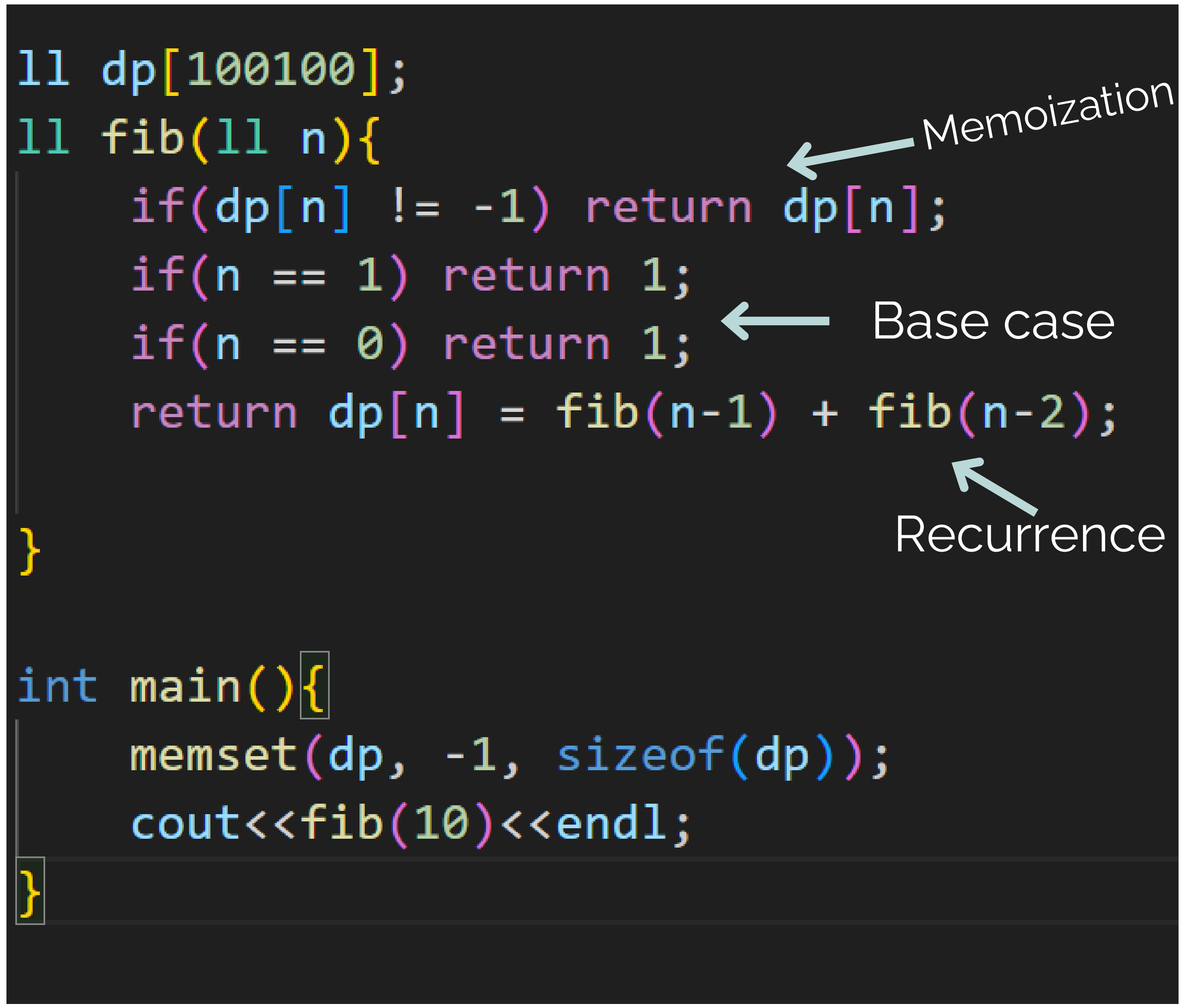
1. What can be the states
2. What my dp will store
3. What are the transitions
4. What are the base cases

Tip 1: First look at the constraints and try to identify the potential states.
Tip 2: Try to break the problem into subproblems to write the transitions/recurrence.

Fibonacci Example

Recursive

Also known as top down approach



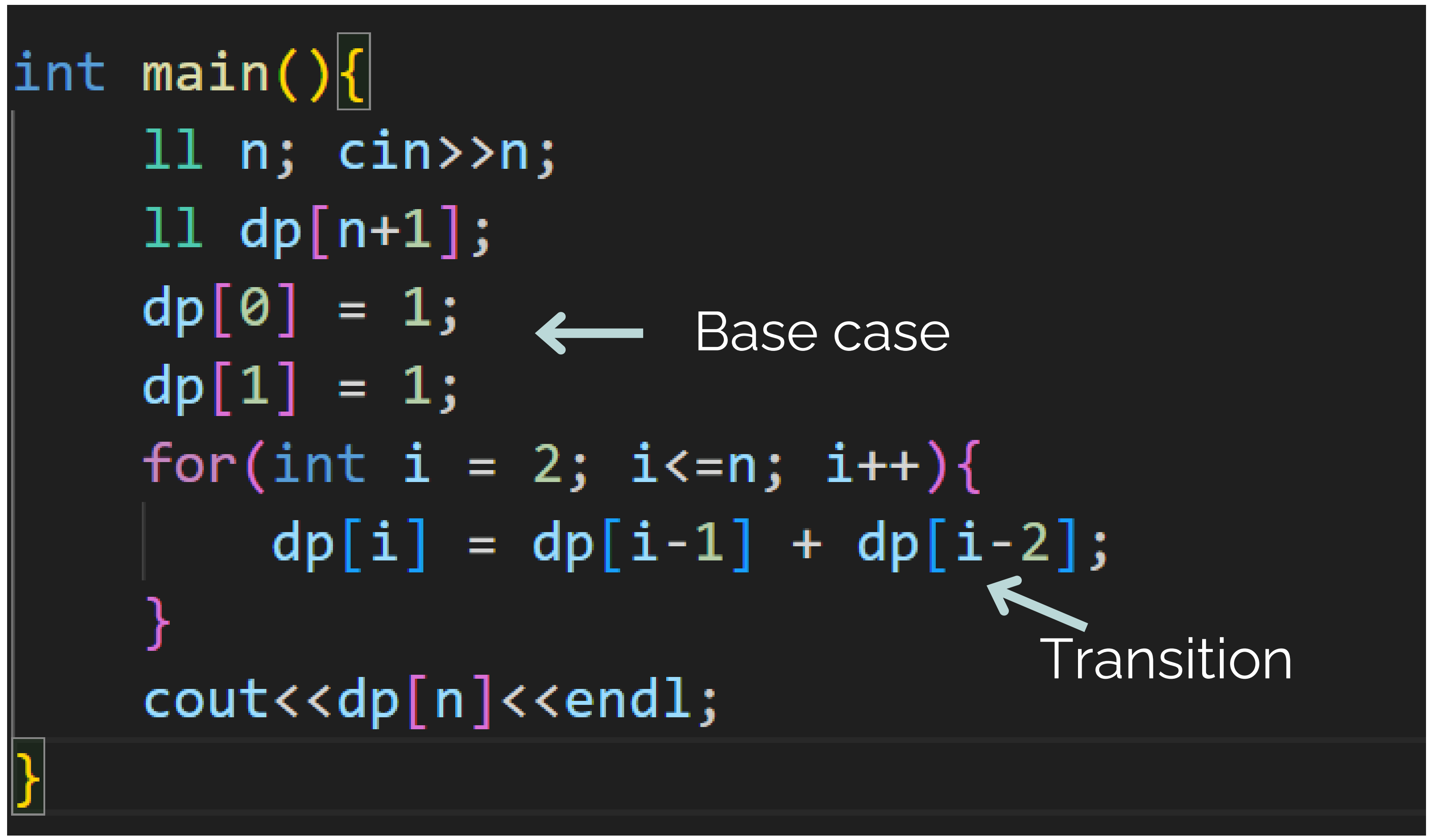
```
ll dp[100100];
ll fib(ll n){
    if(dp[n] != -1) return dp[n];
    if(n == 1) return 1;
    if(n == 0) return 1;
    return dp[n] = fib(n-1) + fib(n-2);

}

int main(){
    memset(dp, -1, sizeof(dp));
    cout<<fib(10)<<endl;
}
```

Iterative

Also known as bottom up or tabulation method



```
int main(){
    ll n; cin>>n;
    ll dp[n+1];
    dp[0] = 1;
    dp[1] = 1;
    for(int i = 2; i<=n; i++){
        dp[i] = dp[i-1] + dp[i-2];
    }
    cout<<dp[n]<<endl;
}
```

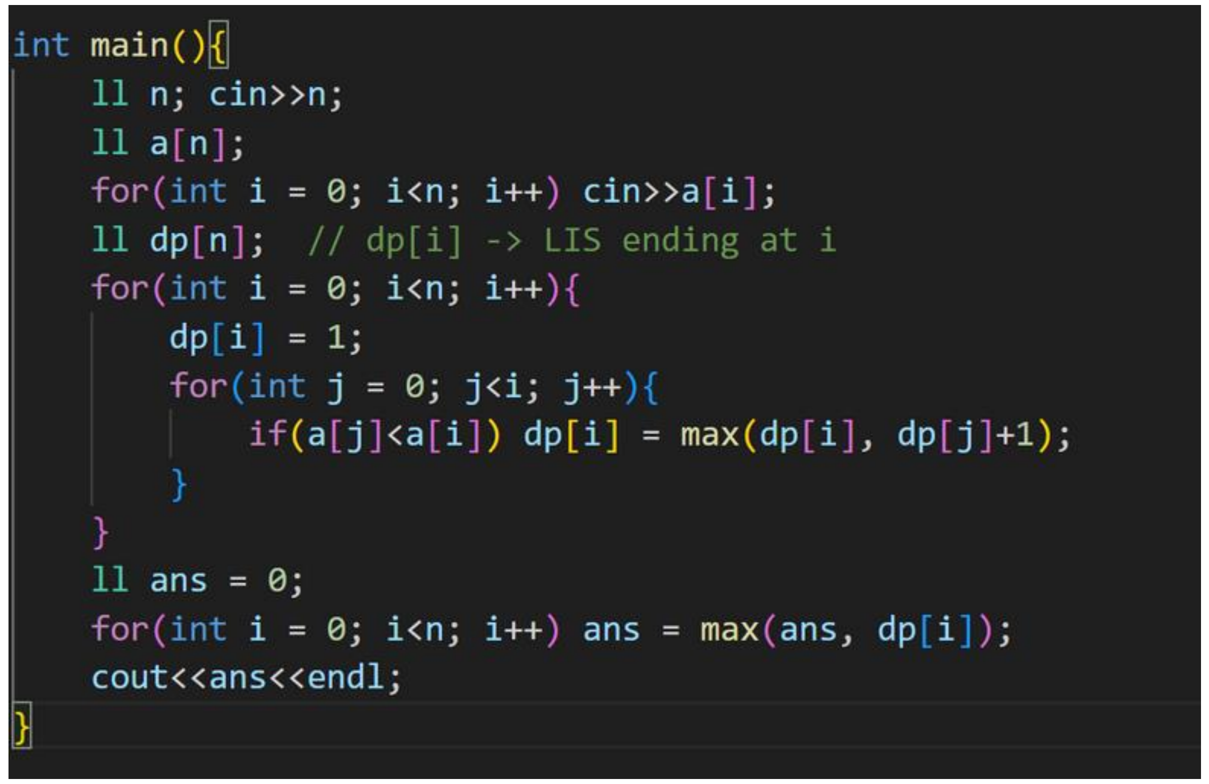
How to calculate time complexity

Time complexity = no of state * avg no of transition

In previous example of fibonacci
no of states = O(n)
no of transition = O(1)
TC = O(n).

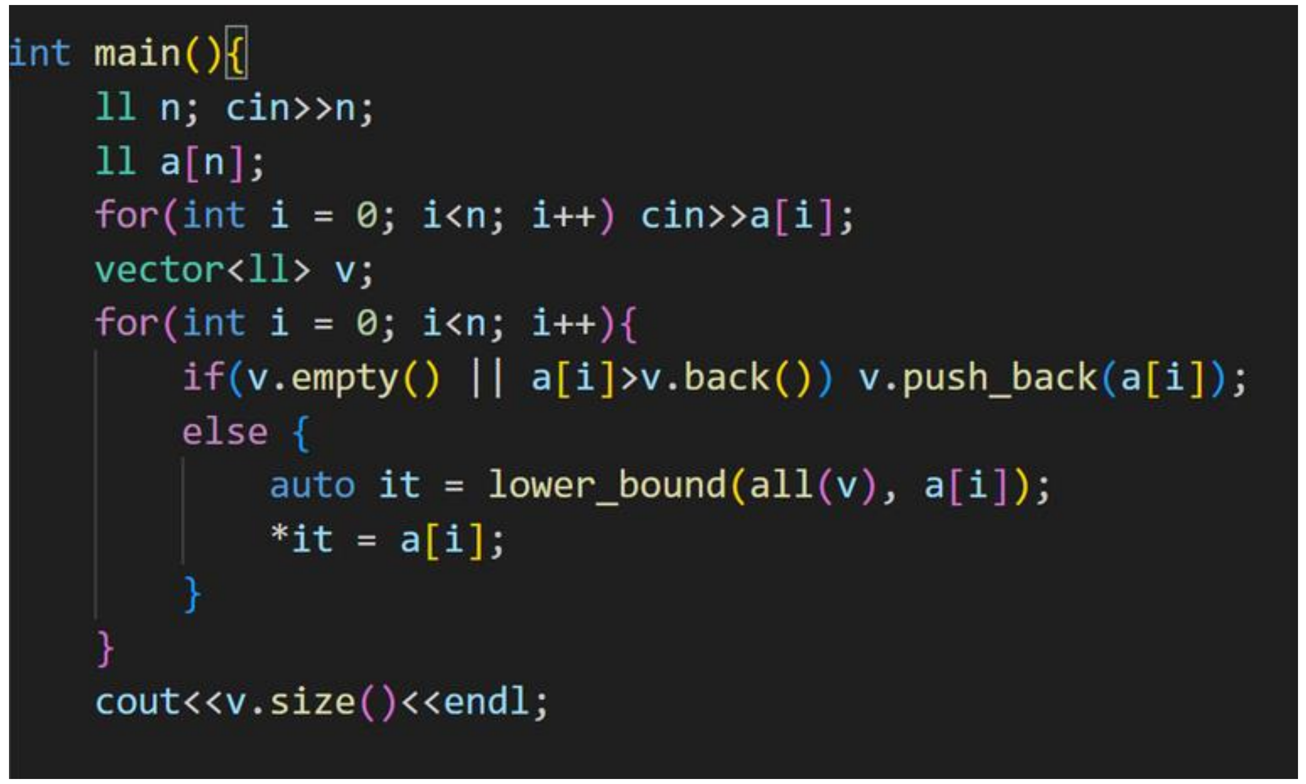
Longest Increasing Subsequence

Shreyans has n friends and his friendship level with each friends is given in the form of an array of integers. He meets them in order from 1 to n and will not meet a friends if his friendship level is less than or equal to the previous friend he met, i.e the next friend he meets must have a friendship level greater than the previous friend he met. Find the maximum no of friends shreyans can meet.



```
int main(){
    ll n; cin>>n;
    ll a[n];
    for(int i = 0; i<n; i++) cin>>a[i];
    ll dp[n];  // dp[i] -> LIS ending at i
    for(int i = 0; i<n; i++){
        dp[i] = 1;
        for(int j = 0; j<i; j++){
            if(a[j]<a[i]) dp[i] = max(dp[i], dp[j]+1);
        }
    }
    ll ans = 0;
    for(int i = 0; i<n; i++) ans = max(ans, dp[i]);
    cout<<ans<<endl;
}
```

Greedy or DP???



```
int main(){
    ll n; cin>>n;
    ll a[n];
    for(int i = 0; i<n; i++) cin>>a[i];
    vector<ll> v;
    for(int i = 0; i<n; i++){
        if(v.empty() || a[i]>v.back()) v.push_back(a[i]);
        else {
            auto it = lower_bound(all(v), a[i]);
            *it = a[i];
        }
    }
    cout<<v.size()<<endl;
```

Printing LIS: https://www.youtube.com/watch?v=XhzQHpGcQg4&t=3s (27:00)

0 or 1

There are n empty positions and you can fill these n positions with 0 or 1. Also no k consecutive positions are same. Find the number of ways you can fill these n positions.

Example: for n = 3 k = 3

- 001
- 011
- 010
- 110
- 100
- 101

Dev's Chocolate

After each exam if Dev's mother gets impressed by his performance, she gives a chocolate packet to him in which there can be any number of pieces of chocolates based on his performance. If Dev did not perform well (which is very unlikely to happen) he doesn't get any chocolate. At the end of all exams Dev got n pieces of chocolate. Find in how many ways he can get those n pieces? (No restriction on number of exams)

Example: for n = 4 there are 5 ways:

- 1 1 1 1
- 1 2 1 (Note: 1 1 2 and 1 2 1 are same)
- 2 2
- 3 1
- 4

https://atcoder.jp/contests/dp/tasks/dp_d

Shashwat’s password

Shashwat has a password in the form of a string of length n. Your task is to crack the password. Since this task is quite difficult, I’ve made it a bit easier for you — the password can be cracked by knowing just a subsequence of the original string (Note: empty string is also a subsequence). So, in how many distinct ways can you crack the password?

Example, abab-> “”, “a”, “b”, “ab”, “aa”, “ba”, “bb”, “aba”, “bab”, “abb”, “aab” , “abab”.

https://codeforces.com/gym/595453/problem/F

Tanmay’s Frustration

After the last lecture on strings, Tanmay was very frustrated. Due to his frustration if he sees a string, he just want to get rid of that string. In one operation he can delete a contiguous substring of the string, if all letters in the substring you delete are equal (Deleting “aaa” from “bcaaacc” gives “bccc”). Also he is very lazy, so he wants to perform as less no of operations as needed. Find the minimum number of operations to delete the complete string.

Example, For s = abaca, min number of operations req = 3

- First remove b, remaining string = aaca
- Then remove ‘c’, remaining string = aaa
- Now remove ‘aaa’, and the string is empty

https://codeforces.com/contest/1132/problem/F

Extra resources

- DP on trees: https://codeforces.com/blog/entry/20935
- Digit Dp: https://codeforces.com/blog/entry/53960
- Bitmask Dp: https://youkn0wwho.academy/topic-list/bitmask_dp
- SOS (sum over subsets) Dp: https://codeforces.com/blog/entry/45223

Questions:

- Atcoder Dp contest: https://atcoder.jp/contests/dp/tasks (Relevant till problem N)
- Cses problem set: https://cses.fi/problemset/ (DP section)

Thank You!