Programming

# Mistakes in Programming

1. Read the input properly (see for input format )
2. Array indexing take care
3. Make perspectives in editor
4. Debug : for debug
5. Explorer : git , explorer and code
6. Code : only code
7. Divide the array properly while doing divide and conquer

int divide(int[] arr , int n){ // n is the array size arr[0 …… n-1]

if(n == 1)

return a[n-1];

mid = n/2 ;

divide( arr , mid);

divide( arr + mid , n – mid)

if ,

n is odd : mid = a[n/2] i.e. for arr[5] mid is arr[2]

n is even : mid = a[n/2] and a[n/2 - 1] i.e. for arr[4] mid is arr[2] and arr[1]

1. Check if the intermediate result exceeds range , then cast to long and recast to int .
2. To reset a vector of vector , initialize inside vector not the vector of vector

i.e.

vector<bool> v\_single(100 , false);

vector<vector<bool> > v\_all (board.size(), v\_single);

1. In backtracking , remember to reset the state if partial solution fails and we are again rebuilding solution .
2. To pass a 2D array in C++ , we can’t pass

Func(int arr[][] , int row , int col )

We must give 2nd dimension of array i.e. arr[][10] ;

Or ,

Use ,

int row = board.size();

int col = board[0].size();

int \* visited = new int[row\*col];

memset(visited, 0, row\*col\*sizeof(int));

for ( int i =0; i< board.size(); i++) {

for ( int j =0; j< board[0].size(); j++) {

visited[i\*col+j] = 1;

1. For using atoi for strings first convert string to char \*

atoi( tokens[i].c\_str() ) ;

1. For Linked List related question , visualize the list while writing code and not only check for links that need to be set but also for links that might accidently get set while setting required links

Recursive solution comes handy while traversing a single LL backwards