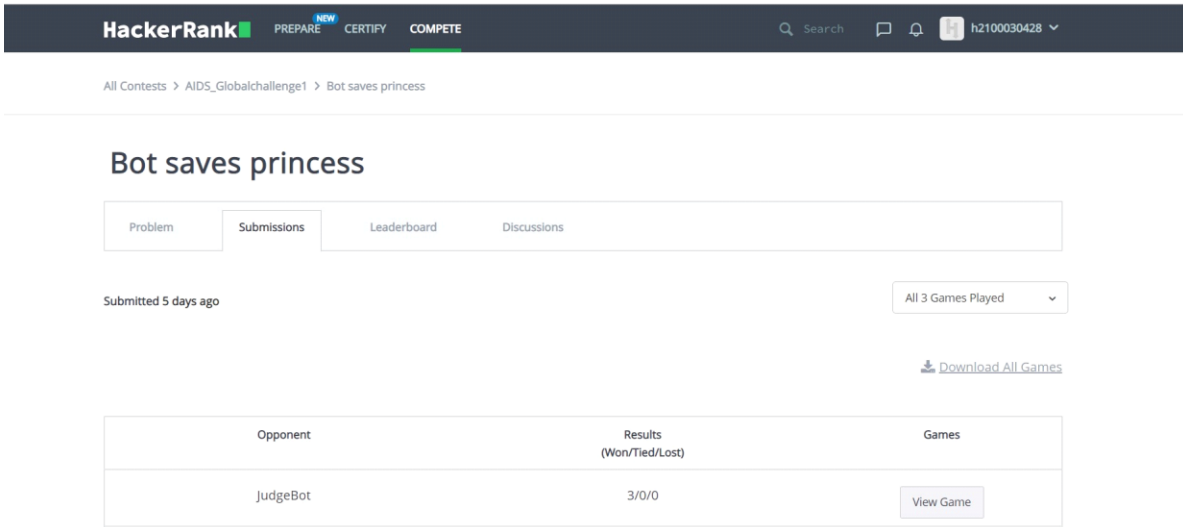
1st Global challenge:

Hackerrank Solutions:

1st sum:

```
#include <stdio.h>

#include <string.h>

#include <math.h>


void displayPathtoPrincess(int n, char grid[101][101]){

   int s=0;

   if(grid[0][0]=='p')

      {

      s=1;

      }

   else if(grid[0][n-1]=='p'){

      s=2;

     }

   else if(grid[n-1][0]=='p')

      { s=3;

      }

   else

      { s=4;

      }

   if(s==1)

      { for(int i=n/2; i>0; i--)

        printf("UP\n");

        for(int j=n/2; j>0; j--)

        printf("LEFT\n");
      }
   else if(s==2)
      { for(int i=n/2; i>0; i--)
        printf("UP\n");
        for(int j=n/2; j>0; j--)
        printf("RIGHT\n");

      }
   else if(s==3)
      { for(int i=n/2; i>0; i--)
        printf("DOWN\n");
        for(int j=n/2; j>0; j--)
        printf("LEFT\n");

      }
   else if(s==4)
      { for(int i=n/2; i>0; i--)
        printf("DOWN\n");
        for(int j=n/2; j>0; j--)
        printf("RIGHT\n");

      }

}
int main(void) {
   int m;
   scanf("%d", &m);
   char grid[101][101]={};
   char line[101];

   for(int i=0; i<m; i++) {
      scanf("%s", line);
      strcpy(grid[i], line);
   }
   displayPathtoPrincess(m,grid);
   return 0;
}
```

OUTPUT:



All Contests > AIDS_Globalchallenge1 > Bot saves princess

Bot saves princess

Submitted 5 days ago

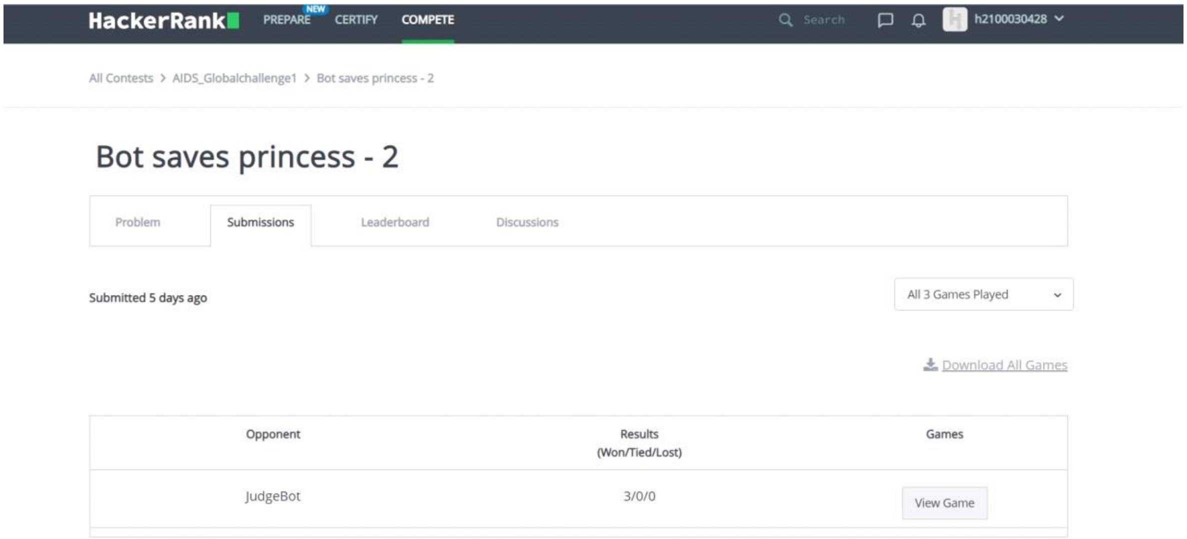
Opponent	Results (Won/Tied/Lost)	Games
JudgeBot	3/0/0	View Game

2nd sum:

```
#include <stdio.h>
```

```
#include <string.h>
#include <math.h>

void nextMove(int n, int r, int c, char grid[101][101]){
  int i, j,a,b;
 short princess[2];


 for (i = 0; i < n; ++i) {
  for (j = 0; j < n; ++j) {
   if (grid[i][j] == 'p') {
    princess[0] = i;
    princess[1] = j;
   }
  }
 }

 if ((a = princess[0] - r) < 0) {
  printf("UP\n");
  return;
 } else if (a > 0) {
  printf("DOWN\n");
  return;
 }


 if ((b = princess[1] - c) < 0) {
  printf("LEFT\n");
  return;
 } else if (b > 0) {
  printf("RIGHT\n");
  return;
 }
}
int main(void) {

  int n, r, c;

  scanf("%d", &n);
  scanf("%d", &r);
  scanf("%d", &c);

           char grid[101][101];

  for(int i=0; i<n; i++) {
    scanf("%s[^\n]%*c", grid[i]);
  }

  nextMove(n, r, c, grid);
  return 0;
}
```

OUTPUT:



Bot saves princess - 2

Submitted 5 days ago

Opponent	Results (Won/Tied/Lost)	Games
JudgeBot	3/0/0	View Game

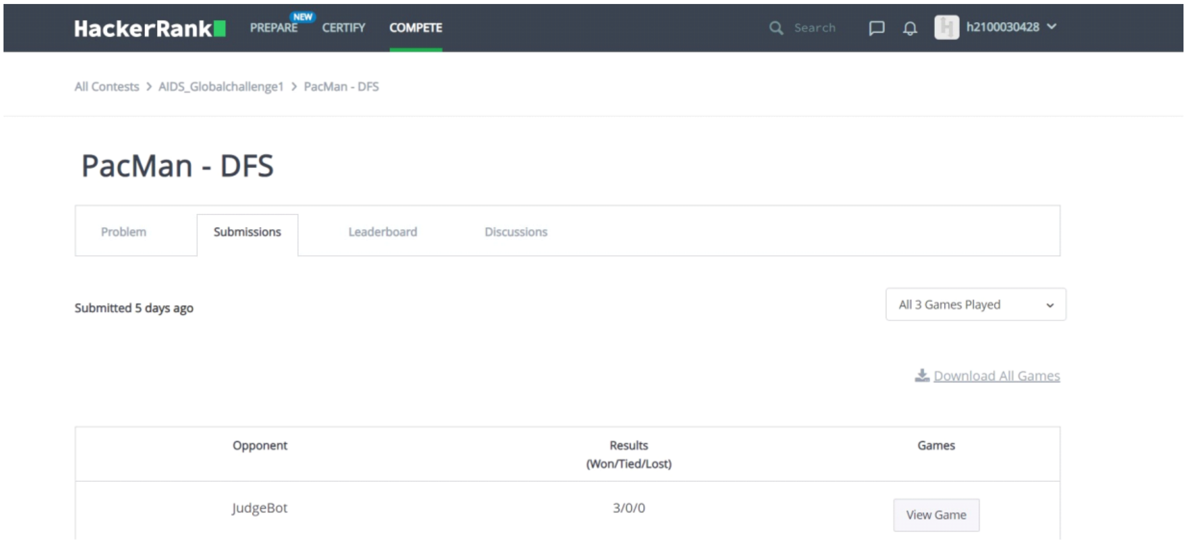
3rd sum:(pacman DFS)

```
import copy

pacman_p, pacman_q = list(map(int, input().split()))

food_p, food_q= list(map(int, input().split()))

n, m = list(map(int, input().split()))

grid = []

node_expanded = []

stack = []

answer_routes = None


for i in range(0, n):

   grid.append(list(map(str, input())))


directions = [[-1, 0], [0, -1], [0, 1], [1, 0]]
```

```
stack.append([pacman_p, pacman_q, []])
while len(stack) > 0:
    p, q, r = stack.pop()
    routes = copy.deepcopy(r)
    routes.append([p, q])

    node_expanded.append([p, q])

    if p == food_p and q == food_q:
        if answer_routes == None:
            answer_routes = routes
            break

    for direction in directions:
        next_p, next_q = p + direction[0], q + direction[1]
        if next_p < 0 or next_p >= n or next_q < 0 and next_q >= n:
            continue

        if grid[next_p][next_q] == "-" or grid[next_p][next_q] == ".":
            grid[next_p][next_q] = '='
            stack.append([next_p, next_q, routes])

print(str(len(node_expanded)))
for point in node_expanded:
    print(str(point[0]) + " " + str(point[1]))

print(str(len(answer_routes) - 1))
```

```
for point in answer_routes:
    print(str(point[0]) + " " + str(point[1]))
```

OUTPUT:



All Contests > AIDS_Globalchallenge1 > PacMan - DFS

PacMan - DFS

Submitted 5 days ago

Opponent	Results (Won/Tied/Lost)	Games
JudgeBot	3/0/0	View Game

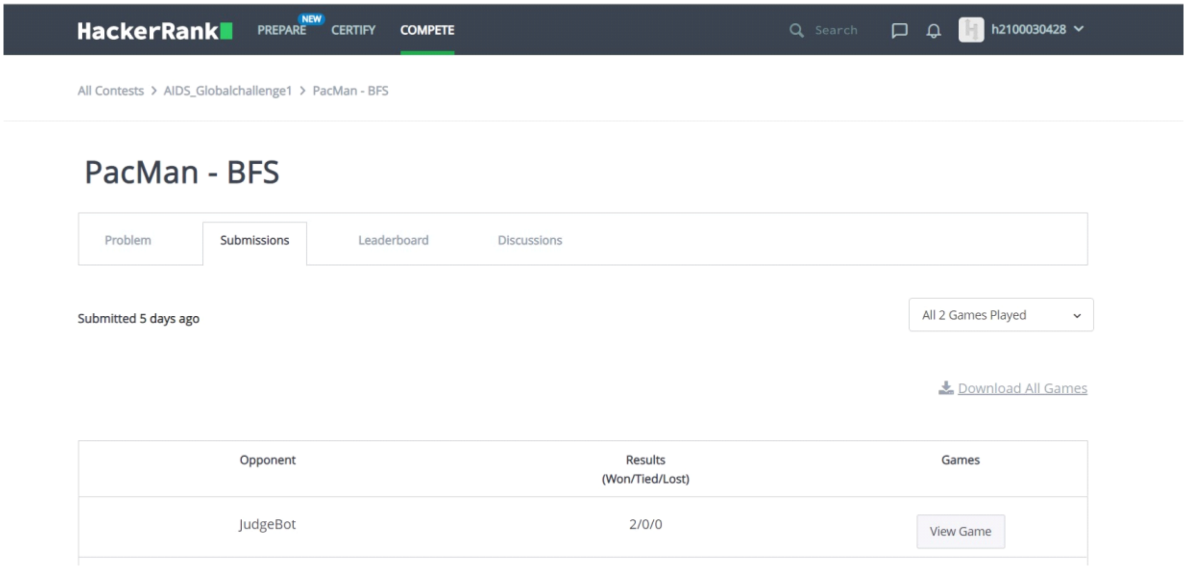
4th sum:(pacman BFS)

```
import copy

pacman_a, pacman_b = list(map(int, input().split()))
food_a, food_b = list(map(int, input().split()))
n, m = list(map(int, input().split()))
grid = []
node_expanded = []
queue = []
answer_routes = None
```

```
for i in range(0, n):
    grid.append(list(map(str, input())))


directions = [[-1, 0], [0, -1], [0, 1], [1, 0]]


queue.append([pacman_a, pacman_b, []])
while len(queue) > 0:
    a, b, r = queue.pop(0)
    routes = copy.deepcopy(r)
    routes.append([a, b])

    node_expanded.append([a, b])

    if a == food_a and b == food_b:
        if answer_routes == None:
            answer_routes = routes
            break

    for direction in directions:
        next_a, next_b = a + direction[0], b + direction[1]
        if next_a < 0 or next_a >= n or next_b < 0 and next_b >= n:
            continue

        if grid[next_a][next_b] == "-" or grid[next_a][next_b] == ".":
            grid[next_a][next_b] = '='
            queue.append([next_a, next_b, routes])
```

```
print(str(len(node_expanded)))
for point in node_expanded:
    print(str(point[0]) + " " + str(point[1]))

print(str(len(answer_routes) - 1))
for point in answer_routes:
    print(str(point[0]) + " " + str(point[1]))
```

OUTPUT:



All Contests > AIDS_Globalchallenge1 > PacMan - BFS

PacMan - BFS

Submitted 5 days ago

Opponent	Results (Won/Tied/Lost)	Games
JudgeBot	2/0/0	View Game

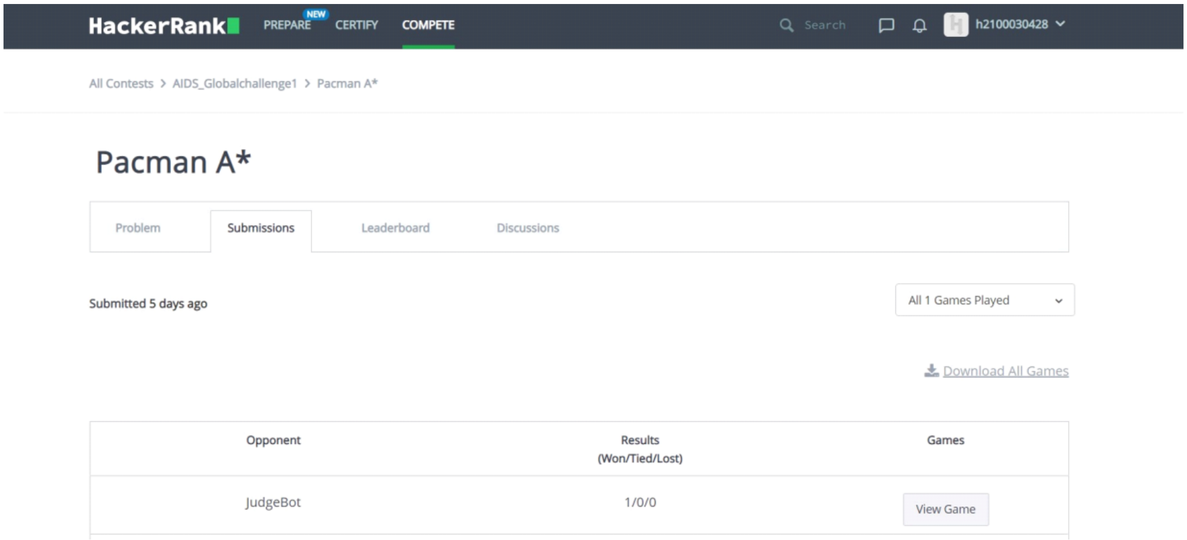
5th sum:(pacman A*)

```
import copy

pacman_a, pacman_b = list(map(int, input().split()))
```

```
food_a, food_b = list(map(int, input().split()))

n, m = list(map(int, input().split()))

grid = []

queue = []

answer_routes = None


for i in range(0, n):

    grid.append(list(map(str, input())))


directions = [[-1, 0], [0, -1], [0, 1], [1, 0]]


queue.append([pacman_a, pacman_b, [], 0])

while len(queue) > 0:

    a, b, p, score = queue.pop(0)

    routes = copy.deepcopy(p)

    routes.append([a, b])


    if a == food_a and b == food_b:

        if answer_routes == None:

            answer_routes = routes

            break


    possible_moves = []

    for direction in directions:

        next_a, next_b = a + direction[0], b + direction[1]

        if next_a < 0 or next_a >= n or next_b < 0 and next_b >= n:

            continue
```

```
      if grid[next_a][next_b] == "-" or grid[next_a][next_b] == ".":
        grid[next_a][next_b] = '='
        possible_moves.append([next_a, next_b, score + abs(food_a - next_a) + abs(food_b - next_b)])

  possible_moves.sort(key = lambda x: x[2])
  for move in possible_moves:
    queue.append([move[0], move[1], routes, score])

print(str(len(answer_routes) - 1))
for point in answer_routes:
  print(str(point[0]) + " " + str(point[1]))
```

OUTPUT:



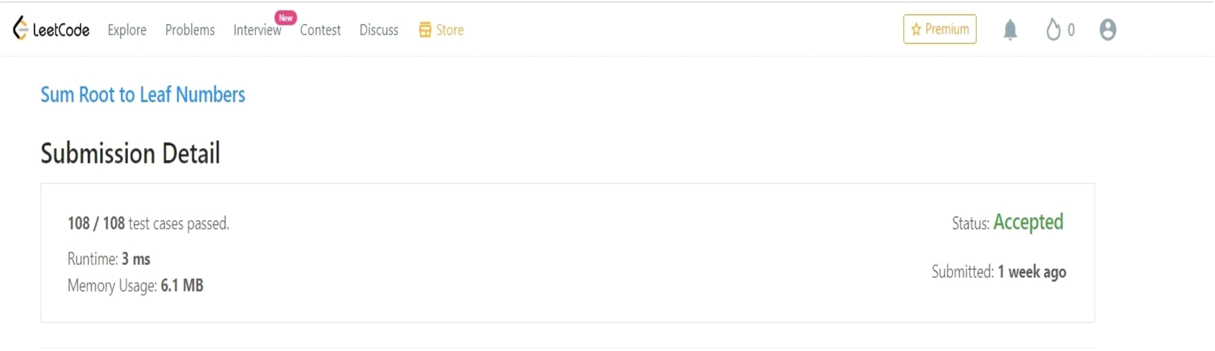
Pacman A*

Submitted 5 days ago

Opponent	Results (Won/Tied/Lost)	Games
JudgeBot	1/0/0	View Game

LEETCODE SOLUTIONS:

1st Sum:



Sum Root to Leaf Numbers

Submission Detail

108 / 108 test cases passed.

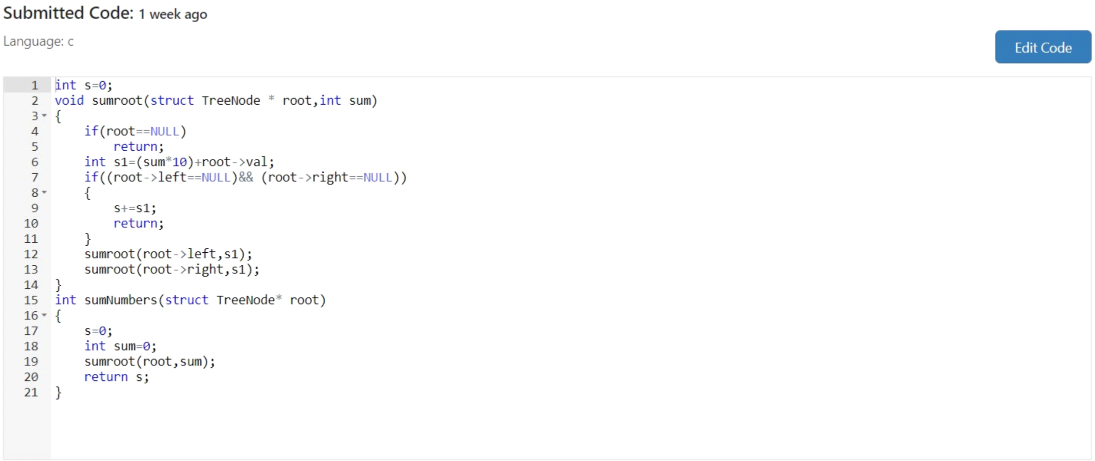
Runtime: **3 ms**

Memory Usage: **6.1 MB**

Status: **Accepted**

Submitted: **1 week ago**

OUTPUT:

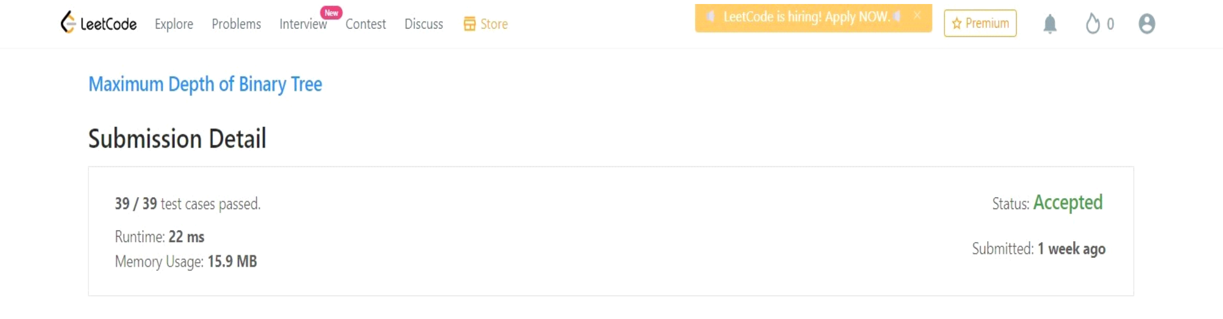


Submitted Code: 1 week ago

Language: c

```c
int s=0;
void sumroot(struct TreeNode * root,int sum)
{
    if(root==NULL)
        return;
    int s1=(sum*10)+root->val;
    if((root->left==NULL)&& (root->right==NULL))
    {
        s+=s1;
        return;
    }
    sumroot(root->left,s1);
    sumroot(root->right,s1);
}
int sumNumbers(struct TreeNode* root)
{
    s=0;
    int sum=0;
    sumroot(root,sum);
    return s;
}
```

2nd Sum:



Maximum Depth of Binary Tree

Submission Detail

39 / 39 test cases passed.

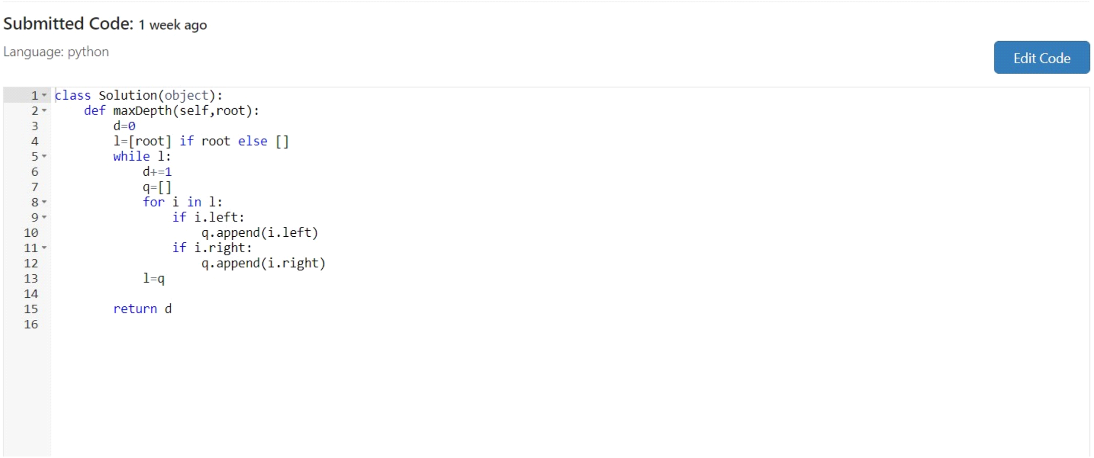
Status: Accepted

Runtime: 22 ms

Memory Usage: 15.9 MB

Submitted: 1 week ago

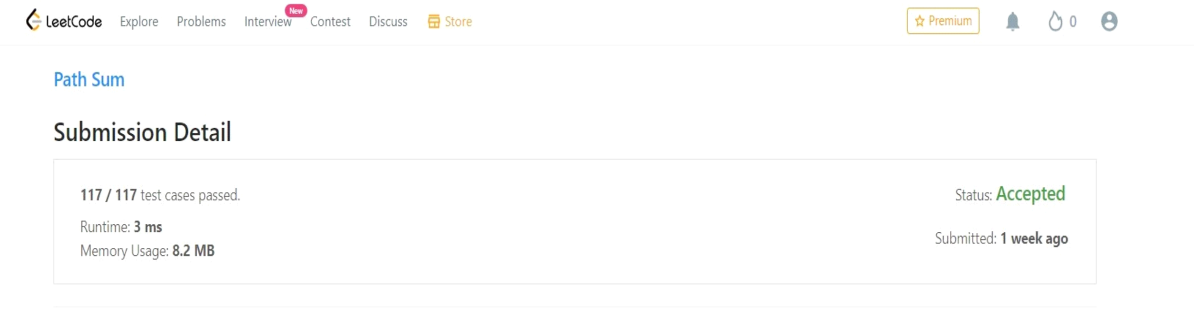
OUTPUT:


Submitted Code: 1 week ago

Language: python

```
class Solution(object):
    def maxDepth(self,root):
        d=0
        l=[root] if root else []
        while l:
            d+=1
            q=[]
            for i in l:
                if i.left:
                    q.append(i.left)
                if i.right:
                    q.append(i.right)
            l=q

        return d

```

3rd Sum:



Path Sum

Submission Detail

117 / 117 test cases passed.

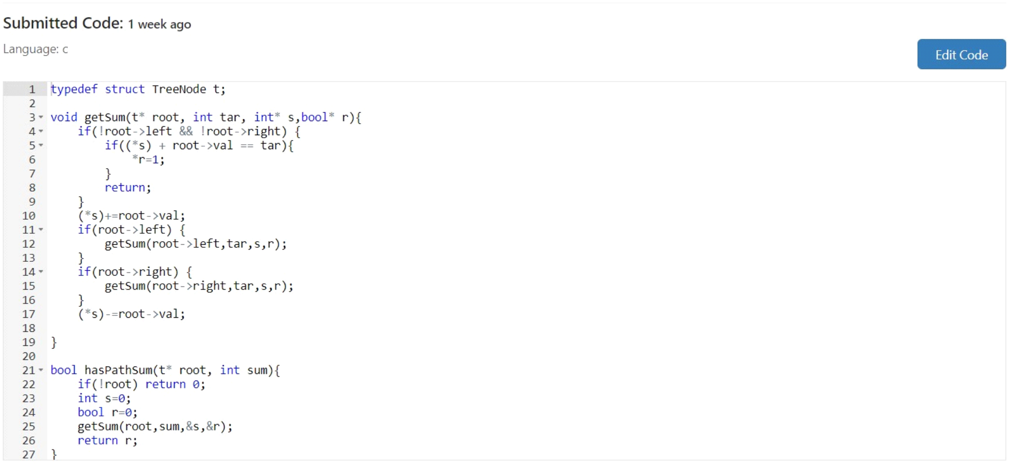
Status: Accepted

Runtime: 3 ms

Memory Usage: 8.2 MB

Submitted: 1 week ago

OUTPUT:

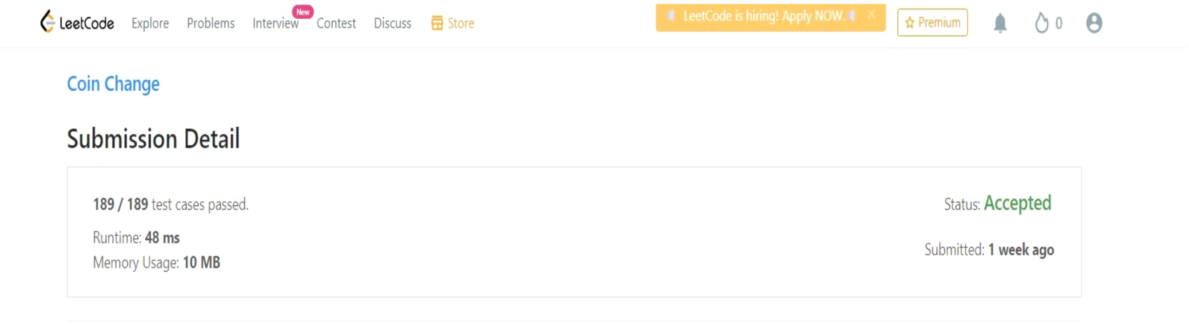


Submitted Code: 1 week ago

Language: c

```c
typedef struct TreeNode t;

void getSum(t* root, int tar, int* s,bool* r){
    if(!root->left && !root->right) {
        if((*s) + root->val == tar){
            *r=1;
        }
        return;
    }
    (*s)+=root->val;
    if(root->left) {
        getSum(root->left,tar,s,r);
    }
    if(root->right) {
        getSum(root->right,tar,s,r);
    }
    (*s)-=root->val;

}

bool hasPathSum(t* root, int sum){
    if(!root) return 0;
    int s=0;
    bool r=0;
    getSum(root,sum,&s,&r);
    return r;
}
```

4th Sum:



Coin Change

Submission Detail

189 / 189 test cases passed.

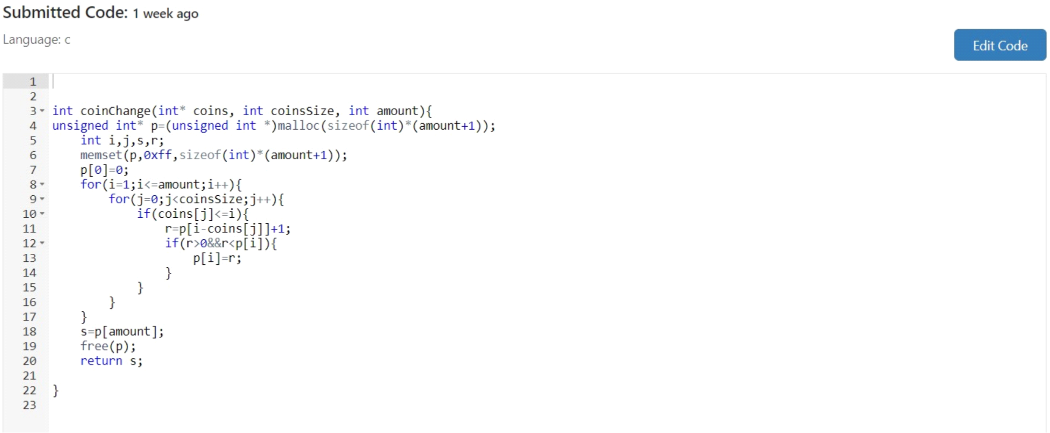
Status: Accepted

Runtime: 48 ms

Memory Usage: 10 MB

Submitted: 1 week ago

OUTPUT:

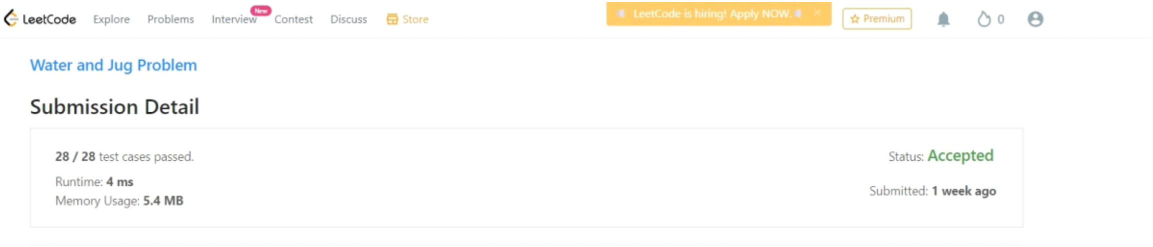


Submitted Code: 1 week ago

Language: c

```c


int coinChange(int* coins, int coinsSize, int amount){
unsigned int* p=(unsigned int *)malloc(sizeof(int)*(amount+1));
    int i,j,s,r;
    memset(p,0xff,sizeof(int)*(amount+1));
    p[0]=0;
    for(i=1;i<=amount;i++){
        for(j=0;j<coinsSize;j++){
            if(coins[j]<=i){
                r=p[i-coins[j]]+1;
                if(r>0&&r<p[i]){
                    p[i]=r;
                }
            }
        }
    }
    s=p[amount];
    free(p);
    return s;

}

```

5th Sum:



Water and Jug Problem

Submission Detail

28 / 28 test cases passed.

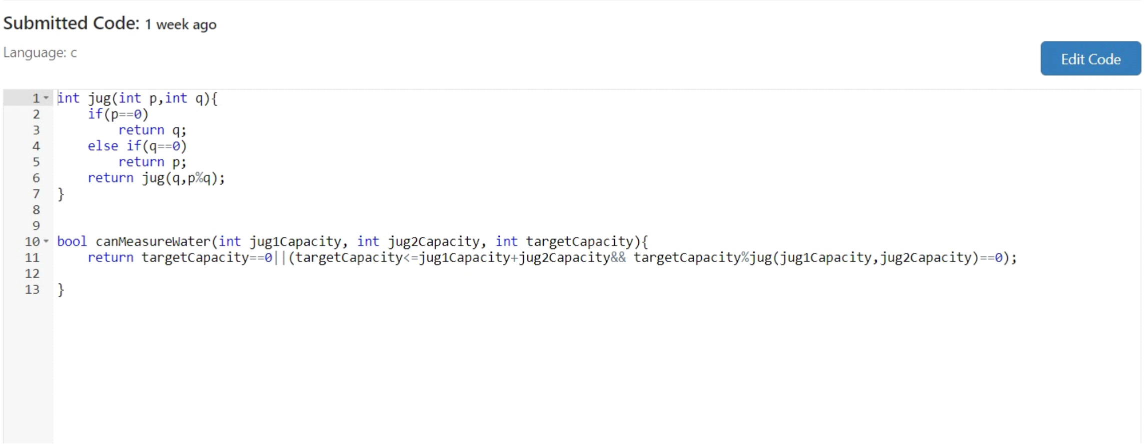
Status: Accepted

Runtime: 4 ms

Memory Usage: 5.4 MB

Submitted: 1 week ago

OUTPUT:



Submitted Code: 1 week ago

Language: c

```c
int jug(int p,int q){
    if(p==0)
        return q;
    else if(q==0)
        return p;
    return jug(q,p%q);
}


bool canMeasureWater(int jug1Capacity, int jug2Capacity, int targetCapacity){
    return targetCapacity==0||(targetCapacity<=jug1Capacity+jug2Capacity&& targetCapacity%jug(jug1Capacity,jug2Capacity)==0);

}
```