LEETCODE QUESTION

Ques 1

class Solution {

public:

vector<int> twoSum(vector<int>& nums, int target) {

int n = nums.size();

for (int i = 0; i < n - 1; i++) {

for (int j = i + 1; j < n; j++) {

if (nums[i] + nums[j] == target) {

return {i, j};

}

}

}

return {};

}

};

Ques 704

class Solution {

public:

int search(vector<int>& nums, int target) {

int low = 0, high = nums.size() - 1;

while (low <= high) {

int mid = low + (high - low) / 2;

if (nums[mid] == target) return mid;

else if (nums[mid] < target) low = mid + 1;

else high = mid - 1;

}

return - 1;

}

};

Ques 287

class Solution {

public:

int findDuplicate(vector<int>& nums) {

int slow = nums[0];

int fast = nums[0];

while (true) {

slow = nums[slow];

fast = nums[nums[fast]];

if (slow == fast) {

break;

}

}

int slow2 = nums[0];

while (slow != slow2) {

slow = nums[slow];

slow2 = nums[slow2];

}

return slow;

}

};

Ques 779

class Solution {

public:

int kthGrammar(int n, int k) {

bool areValuesSame = true;

n = pow(2, n - 1);

while (n != 1) {

n /= 2;

if (k > n) {

k -= n;

areValuesSame = !areValuesSame;

}

}

return ((areValuesSame) ? 0 : 1);

}

};