Vũ Nguyễn Việt Hoàng

Phần 1 : Thuật toán

```
// ex1
function remove_numbers(arr) {
 let rs = []
 let count = {}

 for (let i = 0; i < arr.length; i++) {
   let num = arr[i]
   if (!count[num]) {
     count[num] = 1
   } else {
     count[num]++
   }
   if (count[num] === 1) {
     rs.push(num)
   } else {
     rs.push('*')
   }
 }
 return rs.sort((a, b) => a - b)
}

// ex2
function find_unique_number(arr) {
 let result = 0
 for (let i = 0; i < arr.length; i++) {
   result ^= arr[i]
 }
 return result
}

// ex3
function max_sum(arr) {
 let n = arr.length
 if (n === 0) return 0
 if (n === 1) return arr[0]
 let dp = []
 dp[0] = arr[0]
 dp[1] = Math.max(arr[0], arr[1])
 for (let i = 2; i < n; i++) {
   dp[i] = Math.max(dp[i - 1], dp[i - 2] + arr[i])
 }
 return dp[n - 1]
}
```

```
// ex4
function three_sum(nums) {
 const rs = []

 nums.sort((a, b) => a - b)

 for (let i = 0; i < nums.length - 2; i++) {
   if (i > 0 && nums[i] === nums[i - 1]) continue

   let j = i + 1
   let k = nums.length - 1

   while (j < k) {
     const sum = nums[i] + nums[j] + nums[k]

     if (sum < 0) {
       j++
     } else if (sum > 0) {
       k--
     } else {
       rs.push([nums[i], nums[j], nums[k]])
       while (j < k && nums[j] === nums[j + 1]) j++
       while (j < k && nums[k] === nums[k - 1]) k--
       j++
       k--
     }
   }
 }
 return rs
}

//ex5
function get_height(node) {
 if (node == null) {
   return 0
 }

 const leftHeight = get_height(node.left)
 const rightHeight = get_height(node.right)

 return Math.max(leftHeight, rightHeight) + 1
}
```

Phần 2 : SQL
EX1

1.
```
SELECT email
FROM Canhan
GROUP BY email
HAVING COUNT(email) > 1 ;
```
2.
```
SELECT DISTINCT email FROM Canhan;
```

EX2

```
SELECT salary AS ‘SecondHighestSalary’
FROM luong
ORDER BY salary DESC LIMIT 1,1;
```

EX3

```
SELECT score, DENSE_RANK() OVER(ORDER BY score DESC) AS ‘rank'
FROM ketqua
ORDER BY rank ASC;
```