Cromonix Software Engineer Problem Solving

Time: 3 hrs

Instructions

1. *Create 2 public repositories in your GitHub account. Please share the links when you will finish the tasks.*
    a. *Cromonix_problem_solving_test (Problem 1 & Problem 2 source code should be in this repository)*
    b. *Cromonix_project_test (Problem 3 source code should be in this repository)*
2. *Solve these problems using any programming language.*
3. *Solve the project using any programming language. You can use any framework if you want.*

Expectations

1. *Code & Project should be clean & organized*
2. *Including the test cases will be a plus*
3. *Try to make frequent commits, so that the commit logs show what you have achieved within that time*
4. *Try to use optimized algorithm as much as possible*
5. *Try to avoid copy/paste please*
6. *Variable names, function names, class names should be meaningful*

Problem 1:

Given an sorted array of integers nums and an integer target, return *indices of the two numbers such that they add up to target*.

You may assume that each input would have *exactly* one solution, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Output: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [2,3,4], target = 6
Output: [0,2]
```

Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

Constraints:

- $2 <= nums.length <= 10^8$
- $-10^9 <= nums[i] <= 10^9$
- $-10^9 <= target <= 10^9$
- Only one valid answer exists.

Problem 2:

--- Part One --

Let's say that we found some diagnostic reports of a submarine. We want to find power consumption from these reports.

You need to use the binary numbers in the diagnostic report to generate two new binary numbers (called the gamma rate and the epsilon rate). The power consumption can then be found by multiplying the gamma rate by the epsilon rate.

Each bit in the gamma rate can be determined by finding the most common bit in the corresponding position of all numbers in the diagnostic report. For example, given the following diagnostic report:

00100
11110
10110
10111
10101
01111
00111
11100
10000
11001
00010
01010

Considering only the first bit of each number, there are five 0 bits and seven 1 bits. Since the most common bit is 1, the first bit of the gamma rate is 1.

The most common second bit of the numbers in the diagnostic report is 0, so the second bit of the gamma rate is 0.

The most common value of the third, fourth, and fifth bits are 1, 1, and 0, respectively, and so the final three bits of the gamma rate are 110.

So, the gamma rate is the binary number 10110, or 22 in decimal.

The epsilon rate is calculated in a similar way; rather than use the most common bit, the least common bit from each position is used. So, the epsilon rate is 01001, or 9 in decimal. Multiplying the gamma rate (22) by the epsilon rate (9) produces the power consumption, 198.

Use the binary numbers in your diagnostic report to calculate the gamma rate and epsilon rate, then multiply them together. What is the power consumption of the submarine? (Be sure to represent your answer in decimal, not binary.)

--- Part Two ---

Next, you should verify the life support rating, which can be determined by multiplying the oxygen generator rating by the CO2 scrubber rating.

Both the oxygen generator rating and the CO2 scrubber rating are values that can be found in your diagnostic report - finding them is the tricky part. Both values are located using a similar process that involves filtering out values until only one remains. Before searching for either rating value, start with the full list of binary numbers from your diagnostic report and consider just the first bit of those numbers. Then:

Keep only numbers selected by the bit criteria for the type of rating value for which you are searching. Discard numbers which do not match the bit criteria.

If you only have one number left, stop; this is the rating value for which you are searching.

Otherwise, repeat the process, considering the next bit to the right.

The bit criteria depends on which type of rating value you want to find:

To find oxygen generator rating, determine the most common value (0 or 1) in the current bit position, and keep only numbers with that bit in that position. If 0 and 1 are equally common, keep values with a 1 in the position being considered.

To find CO2 scrubber rating, determine the least common value (0 or 1) in the current bit position, and keep only numbers with that bit in that position. If 0 and 1 are equally common, keep values with a 0 in the position being considered.

For example, to determine the oxygen generator rating value using the same example diagnostic report from above:

Start with all 12 numbers and consider only the first bit of each number. There are more 1 bits (7) than 0 bits (5), so keep only the 7 numbers with a 1 in the first position: 11110, 10110, 10111, 10101, 11100, 10000, and 11001.

Then, consider the second bit of the 7 remaining numbers: there are more 0 bits (4) than 1 bits (3), so keep only the 4 numbers with a 0 in the second position: 10110, 10111, 10101, and 10000.

In the third position, three of the four numbers have a 1, so keep those three: 10110, 10111, and 10101.

In the fourth position, two of the three numbers have a 1, so keep those two: 10110 and 10111.

In the fifth position, there are an equal number of 0 bits and 1 bits (one each). So, to find the oxygen generator rating, keep the number with a 1 in that position: 10111.

As there is only one number left, stop; the oxygen generator rating is 10111, or 23 in decimal.

Then, to determine the CO2 scrubber rating value from the same example above:

Start again with all 12 numbers and consider only the first bit of each number. There are fewer 0 bits (5) than 1 bits (7), so keep only the 5 numbers with a 0 in the first position: 00100, 01111, 00111, 00010, and 01010.

Then, consider the second bit of the 5 remaining numbers: there are fewer 1 bits (2) than 0 bits (3), so keep only the 2 numbers with a 1 in the second position: 01111 and 01010.

In the third position, there are an equal number of 0 bits and 1 bits (one each). So, to find the CO2 scrubber rating, keep the number with a 0 in that position: 01010.

As there is only one number left, stop; the CO2 scrubber rating is 01010, or 10 in decimal.

Finally, to find the life support rating, multiply the oxygen generator rating (23) by the CO2 scrubber rating (10) to get 230.

Use the binary numbers in your diagnostic report to calculate the oxygen generator rating and CO2 scrubber rating, then multiply them together. What is the life support rating of the submarine? (Be sure to represent your answer in decimal, not binary.)

Project

Create a Todo list API application. For success cases use HTTP Status Code 200 and for error cases use HTTP Status Code 400.

Programming Language: Any

Framework: Any

Database: Any

Return type: Json

Endpoints & Description

1. /list
    a. Will return all todo list objects from the database
    b. Pagination should be there
2. /create
    a. Will create a todo list object into the database
    b. Validation should be there.
3. /get/{id}
    a. Retrieve a single todo list object from database
    b. Handle error if the id is not present in the database