Starfleet

Interview

Staff 42 pedago@42.fr

Summary: This document is an interview question for the Starfleet Piscine.

Contents

Chapter I

General rules

- The interview should last between `45 minutes`.
- Both the interviewer and the interviewed student must be present.
- The interviewed student should write his code using a `whiteboard`, with the language of her/his choice.
- At the end of the interview, the interviewer evaluates the student based on the provided criteria.

Read carefully the interview question and solutions, and make sure you `understand` them before the interview. You can't share this document with other students, as they might be interviewed on the same question. Giving them the answer would prevent them from having to solve an unknown question during an interview.

I.1 During the interview

During the interview, we ask you to :

- Make sure the interviewed student `understands` the question.
- Give her/him any `clarification` on the subject that she/he might need.
- Let her/him come up with a solution before you guide her/him to the best solution given the constraints (time and space).
- Ask the student what is the `complexity` of her/his algorithm ? Can it be improved and how ?
- `Guide` her/him to the best solution without giving the answer. You may refer to the `hints` for that.
- You want to evaluate how the interviewed student thinks, so ask her/him to `explain everything` that she/he thinks or writes (there should be no silences).
- If you see a mistake in the code, wait untill the end and give her/him a chance to correct it by her/himself.
- Ask the student to show how the algorithm works on an `example`.
- Ask the student to explain how `limit cases` are handled.
- Bring out to the student any mistake she/he might have done.
- Give `feedback` on her/his performances after the interview.
- Be `fair` in your evaluation.

As always, stay mannerly, polite, respectful and constructive during the interview. If the interview is carried out smoothly, you will both benefit from it !

Chapter II

Multiplication

II.1 Interview question

Given an array of integers, multiply all fields except it's own position.

Return the result in a new array.

Example :

```
Input  : {2, 3, 1, 4}
Output : {12, 8, 24, 6}
```

II.2 Acceptable answers (no constraint)

II.2.1 Brute force

The simplest solution is to compute, for each element of the array, the product of all the other elements.

```
O(n^2) time , O(n) space
where n is the number of elements in the array
```

Note : O(n) space corresponds to the returned array.

code:

```
int multiplyExcept(int *arr, int n, int i) {
        int p = 1;

        for (int j = 0; j < n; j++) {
                if (j != i)
                        p *= arr[j];
        }
        return (p);
}

int *multiply(int *arr, int n) {
        int *res = NULL;

        if (!(res = (int *)malloc(sizeof(int) * n)))
                return (NULL);
        for (int i = 0; i < n; i++)
                res[i] = multiplyExcept(arr, n, i);
        return (res);
}
```

II.2.2 With division

A better solution is to compute the product of all the elements in the array and then fill the new array with the `division` of that product by the value of the current element.

```
O(n) time , O(n) space
where n is the number of elements in the array
```

code:

```
int *multiply(int *arr, int n) {
        int *res = NULL;
        int product = 1;

        for (int i = 0; i < n; i++)
                product *= arr[i];
        if (!(res = (int *)malloc(sizeof(int) * n)))
                return (NULL);
        for (int i = 0; i < n; i++)
                res[i] = product / arr[i];
        return (res);
}
```

II.3 Follow up question

You are not allowed to use `division`.

Your algorithm must run in `O(n) time`.

II.3.1 Hints

- For each element, the result is the multiplication of the product of all the elements on the left by the product of all the elements on the right.
- Can you do it in 2 steps?

II.4 Best solutions

II.4.1 Without division in 2 steps

A good solution is to run through the array twice :

- The first time to get the product of all the elements on the left.
- The second time to get the product of all the elements on the right.

Then, we just have to multiply these two products.

```
O(n) time , O(n) space
where n is the number of elements in the array
```

`Note` : The final product can be obtained directly in the second step (as shown by the code below).

code:

```c
int *multiply(int *arr, int n) {
        int *res = NULL;
        int p = 1;

        if (!(res = (int *)malloc(sizeof(int) * n)))
                return (NULL);
        for (int i = 0; i < n; i++) {
                res[i] = p;
                p *=  arr[i];
        }
        p = 1;
        for (int i = n - 1; i >= 0; i--) {
                res[i] *= p;
                p *= arr[i];
        }
        return (res);
}
```