27 Nov, 2017   Helpful (13)

Front End Engineer Interview

Anonymous Interview Candidate in Menlo Park, CA (US)

🟥 No Offer   🟩 Positive Experience   🟨 Average Interview

Application

I applied through a recruiter. The process took 2 weeks. I interviewed at Facebook (Menlo Park, CA (US)) in August-2017.

Interview

Skype interview

Interview Questions
Given input:

```
// could be potentially more than 3 keys in the object above
items = [
{color: 'red', type: 'tv', age: 18},
{color: 'silver', type: 'phone', age: 20}
...
]

excludes = [
{k: 'color', v: 'silver'},
{k: 'type', v: 'tv'},
....
]
function excludeItems(items, excludes) {
  excludes.forEach(pair => {
    items = items.filter(item => item[pair.k] === item[pair.v]);
  });
  return items;
}
```

1. Describe what this function is doing...
2. What is wrong with that function ?
3. How would you optimize it ?
└▸   12 Answers

❬f❭ ❬t❭ ❬✉❭ ❬🔗❭   **Helpful (13)**   🏳

---

17 Nov, 2016   Helpful (25)

Front End Engineer Interview

Anonymous Interview Candidate in Menlo Park, CA (US)

🟥 No Offer   🟩 Positive Experience   🟥 Difficult Interview

Application

I applied through a recruiter. The process took 2+ months. I interviewed at Facebook (Menlo Park, CA (US)) in October-2016.

Interview

I applied through a recruiter who contacted me in September. She was nice and we booked a Phone/Screen coding interview.

They are using CoderPad for these. This first interview consisted of 3 questions, they all seemed pretty easy and then the interviewer asked for edge cases, a bit more challenging but nothing

```javascript
const items = [ { color: 'red', type: 'tv', age: 18 } ];
const excludes = [ { k: 'color', v: 'blue' }, { k: ..., v: ... } ];
const exclude = (items, excludes) => {
    const query = excludes.reduce(
            (acc, cur) => {
                const { k, v } = cur;
                if (!(k in acc)) acc[k] = new Set();
                acc[k].add(v);
                return acc;                    { color: [ 'blue' ],
            },                                   type: [ ... ] }
            {})
        );
    return items.filter(item => {
        for (const key in query) {
            if ((key in item) && query[key].has(item[key])) {
                return false;
            }
        }
        return true;
    }
}
```