

Coding Challenge - Users Hierarchy

In our system each user belongs to a user-group with a defined set of permissions.

We name such a group "Role". A certain role (unless it is the root) must have a parent role to whom it reports to. For example a customer may have these roles in their account:

```
objRole1 = {
    "Id": 1,
    "Name": "System Administrator",
    "Parent": 0
};

objRole2 = {
    "Id": 2,
    "Name": "Location Manager",
    "Parent": 1
};

objRole3 = {
    "Id": 3,
    "Name": "Supervisor",
    "Parent": 2
};

objRole4 = {
    "Id": 4,
    "Name": "Employee",
    "Parent": 3
};

objRole5 = {
    "Id": 5,
    "Name": "Trainer",
    "Parent": 3
};
```

Notice how the System Administrator has no parent role and how Employee has as parent role the Supervisor.

Naturally this cascading parent-child relationship means that Location Manager, Supervisor, Employee, Trainer are all children roles to System Administrator.

Some users in that account may look as follows:

```
objUser1 = {
    "Id": 1,
```

```

        "Name": "Adam Admin",
        "Role": 1
    };

objUser2 = {
    "Id": 2,
    "Name": "Emily Employee",
    "Role": 4
};

objUser2 = {
    "Id": 3,
    "Name": "Sam Supervisor",
    "Role": 3
};

objUser3 = {
    "Id": 4,
    "Name": "Mary Manager",
    "Role": 2
};

objUser4 = {
    "Id": 5,
    "Name": "Steve Trainer",
    "Role": 5
};

```

Task

Come up with a function, for an arbitrary collection of roles and users, given a user Id returns a list of ALL their subordinates (i.e: including their subordinate's subordinates). For example if you were given user #3 in the above example (Sam Supervisor), you should output objUser2 (Emily Employee) and objUser5 (Steve Trainer)

Another example is if you were give user #1 in the above example (Adam Admin), you should output a list containing [objUser2, objUser3, objUser4, objUser5] in no particular order.

Sample Input

```

roles = [
    {
        "Id": 1,
        "Name": "System Administrator",
        "Parent": 0
    }
]

```

```

    },
    {
      "Id": 2,
      "Name": "Location Manager",
      "Parent": 1
    },
    {
      "Id": 3,
      "Name": "Supervisor",
      "Parent": 2
    },
    {
      "Id": 4,
      "Name": "Employee",
      "Parent": 3
    },
    {
      "Id": 5,
      "Name": "Trainer",
      "Parent": 3
    }
  ]
};

```

```

users = [
  {
    "Id": 1,
    "Name": "Adam Admin",
    "Role": 1
  },
  {
    "Id": 2,
    "Name": "Emily Employee",
    "Role": 4
  },
  {
    "Id": 3,
    "Name": "Sam Supervisor",
    "Role": 3
  },
  {
    "Id": 4,
    "Name": "Mary Manager",
    "Role": 2
  },
  {

```

```

        "Id": 5,
        "Name": "Steve Trainer",
        "Role": 5
    }
];

setRoles(roles);
setUsers(users);
getSubOrdinates(3); // should return [{"Id": 2, "Name": "Emily
Employee", "Role": 4}, {"Id": 5, "Name": "Steve Trainer", "Role": 5}]
getSubOrdinates(1); // should return [{"Id": 2, "Name": "Emily
Employee", "Role": 4}, {"Id": 3, "Name": "Sam Supervisor", "Role": 3},
{"Id": 4, "Name": "Mary Manager", "Role": 2}, {"Id": 5, "Name": "Steve
Trainer", "Role": 5}]

```

Ground rules

- Package your solution in any way you would like (e.g: zip file, github repo, etc)
 - Include a README.md with your solution to tell us how to get it running •
- Produce a test suite that we can run with everything passing
- Make sure you write readable code.
 - Feel free to write comments explaining your solution so we understand your thinking behind it
 - Email the solution to your recruiter. They will forward it to the engineers involved in your role
 - Have fun!