

Take Home Assignment (JavaScript)

Requirements

Given an array of clicks, return the subset of clicks where:

1. For each IP within each one hour period, only the most expensive click is placed into the result set.
2. If more than one click from the same IP ties for the most expensive click in a one hour period, only place the earliest click into the result set.
3. If there are more than 10 clicks for an IP in the overall array of clicks, do not include any of those clicks in the result set.

The result set should be stored in an array of objects. Each object should represent a click. The expected result set should be a subset of the original array.

Definitions

1. A **click** is the composite of an IP address, a timestamp, and a click amount.
2. **Duplicate clicks** are clicks that have the same IP address, regardless of timestamp or click amount.
3. **Click periods** are defined as the one hour spans that start at the top of the hour. So, in one day, there are 24 periods and they are broken down as follows (in HH:MM:SS format):

00:00:00 - 00:59:59 (period 1)

01:00:00 - 01:59:59 (period 2)

02:00:00 - 02:59:59 (period 3)

...

22:00:00 - 22:59:59 (period 23)

23:00:00 - 23:59:59 (period 24)

The Array of Clicks

Please see the attached **clicks.json** file.

Submitting Code

Please reply to the sender of the assignment and attach a solution_<firstName>_<lastName>.zip archive which includes the following:

- A well written and performant solution, implemented in JavaScript (ES5 or ES6 are fine, TypeScript and CoffeeScript are not allowed), that will run when the command ``npm run solution`` is executed
- **result-set.json** - A json file with the result set produced by your solution
- A thorough suite of tests that will run when the command ``npm run test`` is executed
- **README.md** - A readme that contains any pertinent information
- You can use whatever libraries you would like, but they should install with ``npm install``. Please explain and justify your library choices in **README.md**

Grading criteria:

- **Correctness:** solutions must produce correct result sets given any valid input
- **Readability & Organization:** code should be well organized and readable with well thought out architecture and naming choices
- **Performance:** the algorithm implemented by your solution should be as performant as possible