Question #1: Counting Website Visitors

Task

Your website is getting popular and you'd like to get more insights about your Traffic. You want to know how many visits, unique visitors and sessions you are serving.

Implement a solution for this in:

`com.squarespace.interview.visitors.AnalyzeVisitorsSolution#processPageViews()`

Definitions

Visits refer to the number of times a site is visited, no matter how many visitors make up those views.

Unique visitors refer to the number of distinct individuals requesting pages from the website, regardless of how often they view it.

Sessions refer to the time a visitor is active on a website. It is defined as a 30min sliding window.

- So for example, if a visitor views your site every 10min for 50min, then all those views are part of the same session (6 views, 1 unique visitor, 1 session).
- But if a visitor is inactive for more than 30min, and they view your site again a little bit later, that creates a new session for this visitor.

Input

You will receive an array of WebsiteVisit objects. These are ordered chronologically (from oldest to most recent), and are defined as follows:

```
public class WebsiteVisit {
    private final String visitorId;
    private final long timestamp;
}
```

The timestamp field is measured in seconds.

Output

You will return a array of long integers in this order:

- count of site visits
- count ofunique visitors
- count of sessions

Question #2: Finding a point in a hierarchy of views

You are working with a UI template framework that represents a page as a tree of rectangular nodes. Rectangles nest within each other to form a complex rendering of rectangles. They can nest several levels down.

This uses a 2d coordinate system where the upper-left corner has the coordinates `(0,0)`. The x axis increases towards the right and the y increases towards the bottom. It is a zero-based scale, so a pixel at `(10,10)` would be outside a square with width/height of 10.

Task

You will receive a `Node` and a `java.awt.Point`. The node represents the root rectangle in the tree of nodes, and your task is to return an ordered list of nodeIds representing the path to the given point in the tree of nodes.

Implement a solution for this in:

`com.squarespace.interview.findpoint.FindPointSolution#findPathToNode()`

Input

`Node` class:

```
class Node {
  private final String id;
  private final Integer left;
  private final Integer top;
  private final Integer width;
  private final Integer height;
  private final List<Node> children;
}
```

Children are strictly contained in their parent (no overflow). This means that:

```
0 < child.left + child.width <= parent.width
```

```
0 < child.top + child.height <= parent.height
```

The origin of a given node is relative to the origin of its parent.

Children nodes may overlap each other. In this case, the child with the higher index covers the child with the lower index. Children may not completely fill a parent. The `id` of a node is an opaque string with no intrinsic significance.

Output

Return a `List<String>` describing the path from the root `Node` to the `Node` that you found. If there is no match, return an empty array.

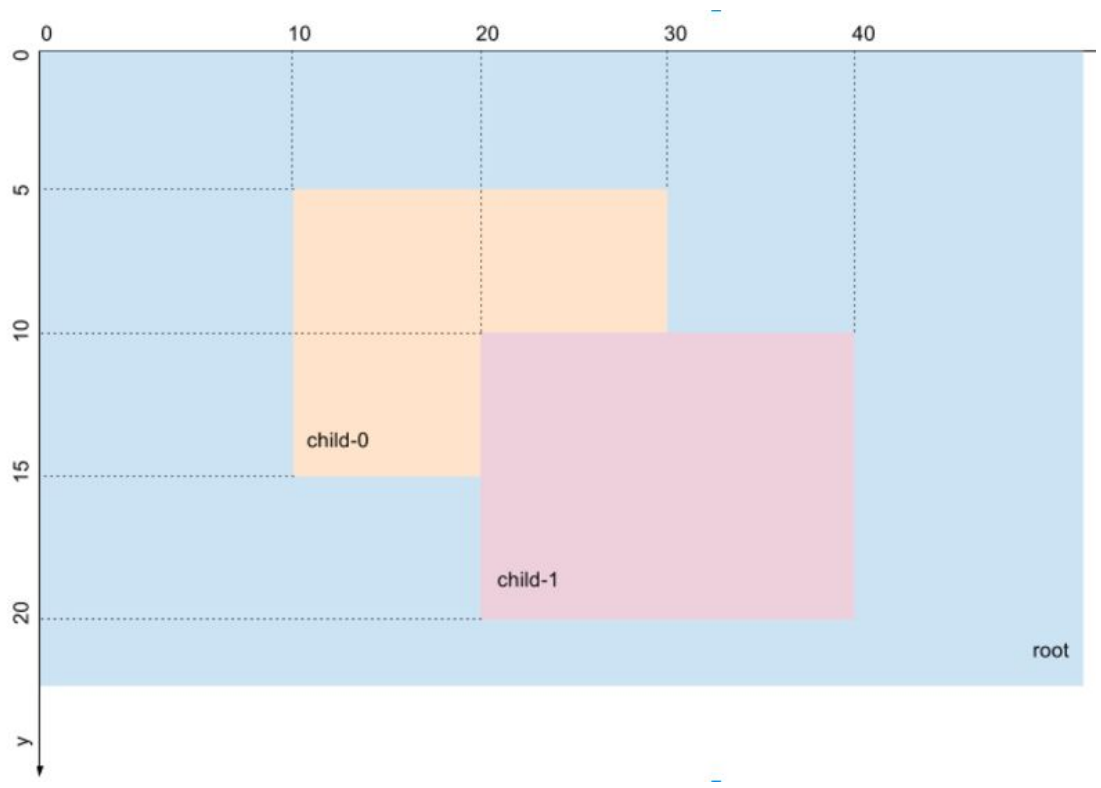
Example

Note that if 2 children overlap, the one with the highest index in the children array is chosen. Also, keep in mind that a child node can have child nodes of their own.

For example, given this JSON describing a hierarchy:

```
{
  "id": "root",
  "left": 0, "top": 0,
  "width": 54, "height": 23,
  "children": [
    {
      "id": "child-0",
      "left": 10, "top": 5,
      "width": 20, "height": 10,
      "children": []
    },
    {
      "id": "child-1",
      "left": 20, "top": 10,
      "width": 20, "height": 10,
      "children": []
    }
  ]
}
```

It describes an hierarchy like this:



In this example, for a pixel with coordinate `(22, 12)`, the pixel falls into the bounds of `child-0` and `child-1` but `child-1` appears last in the children array. Therefore, you should thus return the path from `root` to `child-1` -> `["root", "child-1"]`.