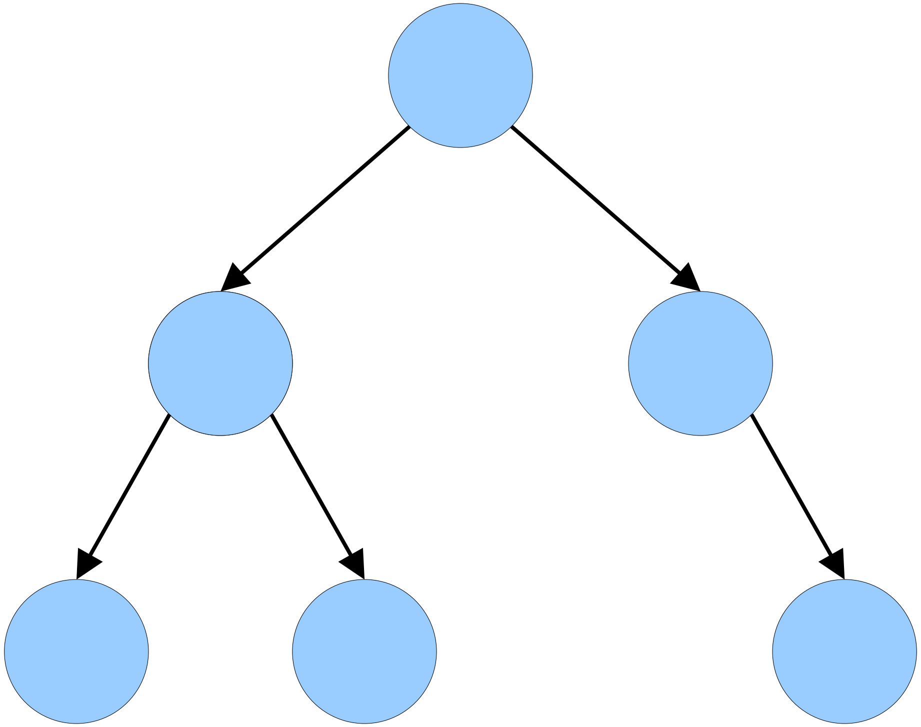
# Multiway Trees

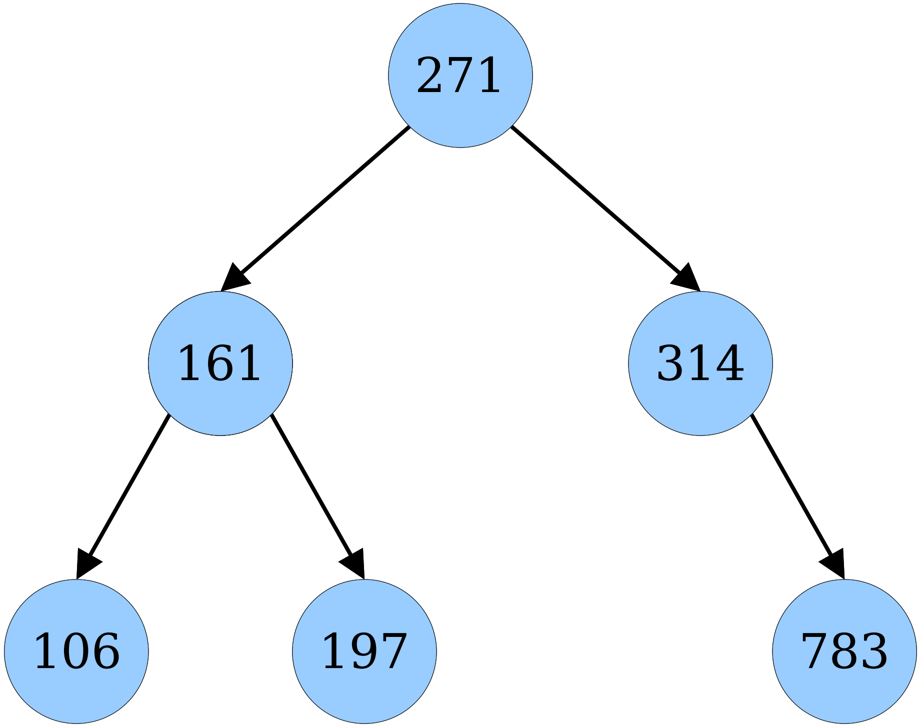
# Outline for Today

- ***Multiway Trees***
  - Expanding our tree possibilities.
- ***Representing Multiway Trees***
  - Modeling branching in software.
- ***Working With Real-World Data***
  - JSON and public data sets.

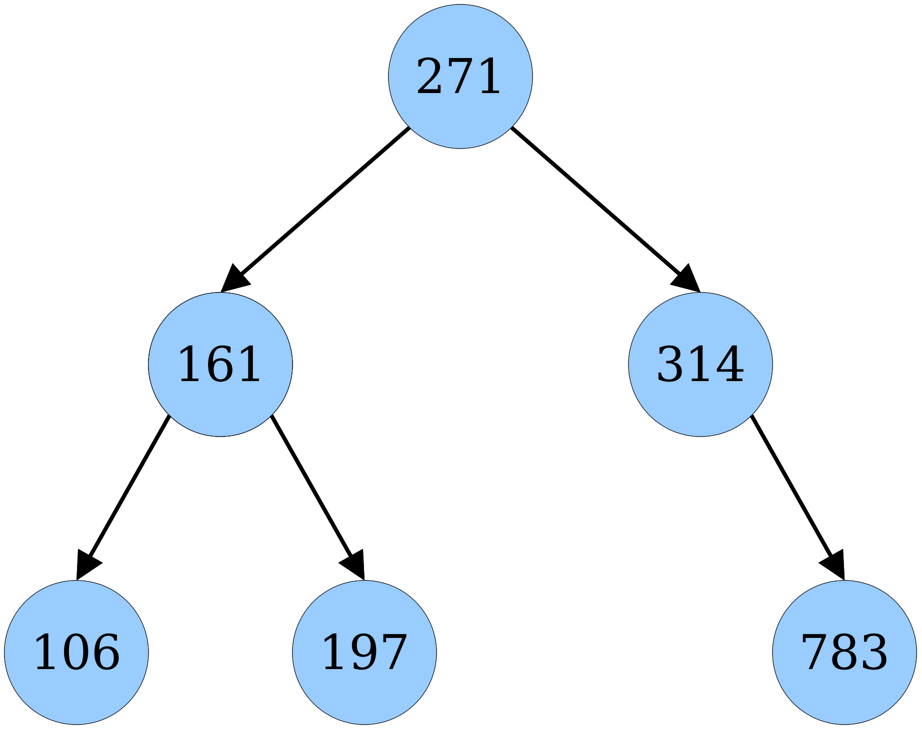
# Binary Search Trees



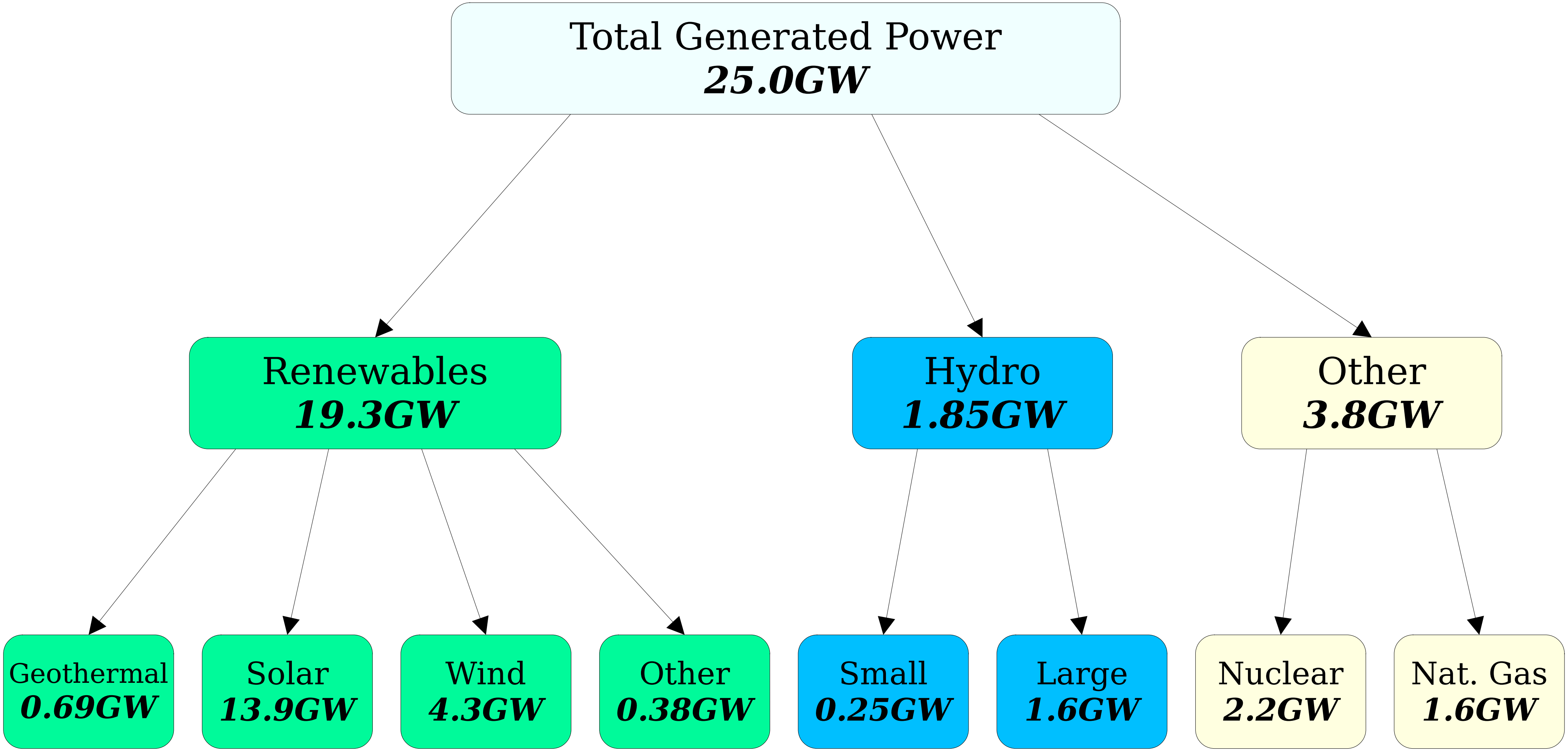
# Binary Search Trees



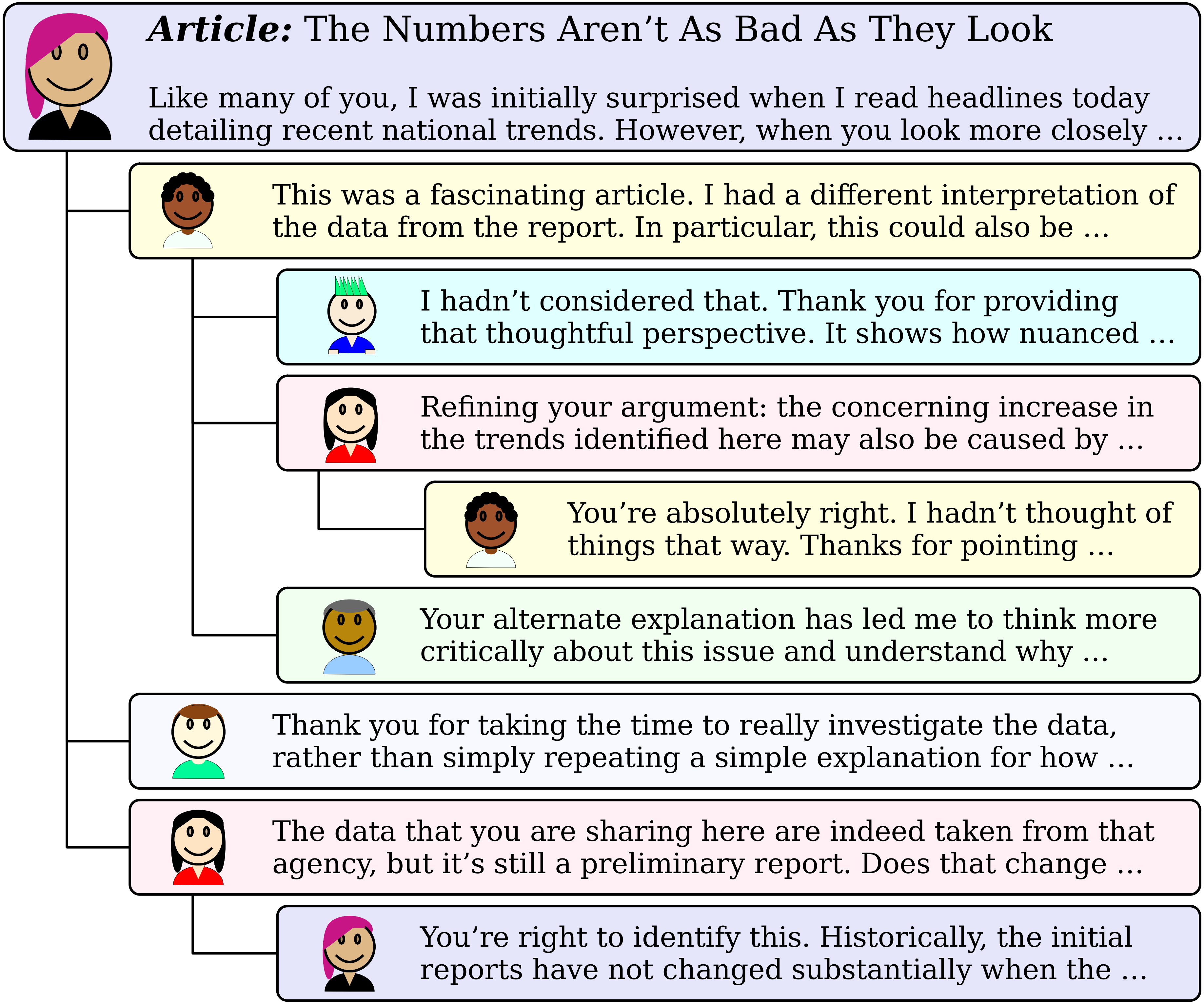
Binary Search Trees



# Binary Search Trees



*Electric Power Generation in California, March 2, 2025, 8:20AM*
*Source: California Independent System Operator (CAISO)*



***Article:*** The Numbers Aren’t As Bad As They Look

Like many of you, I was initially surprised when I read headlines today detailing recent national trends. However, when you look more closely …

This was a fascinating article. I had a different interpretation of the data from the report. In particular, this could also be …

I hadn’t considered that. Thank you for providing that thoughtful perspective. It shows how nuanced …

Refining your argument: the concerning increase in the trends identified here may also be caused by …

You’re absolutely right. I hadn’t thought of things that way. Thanks for pointing …

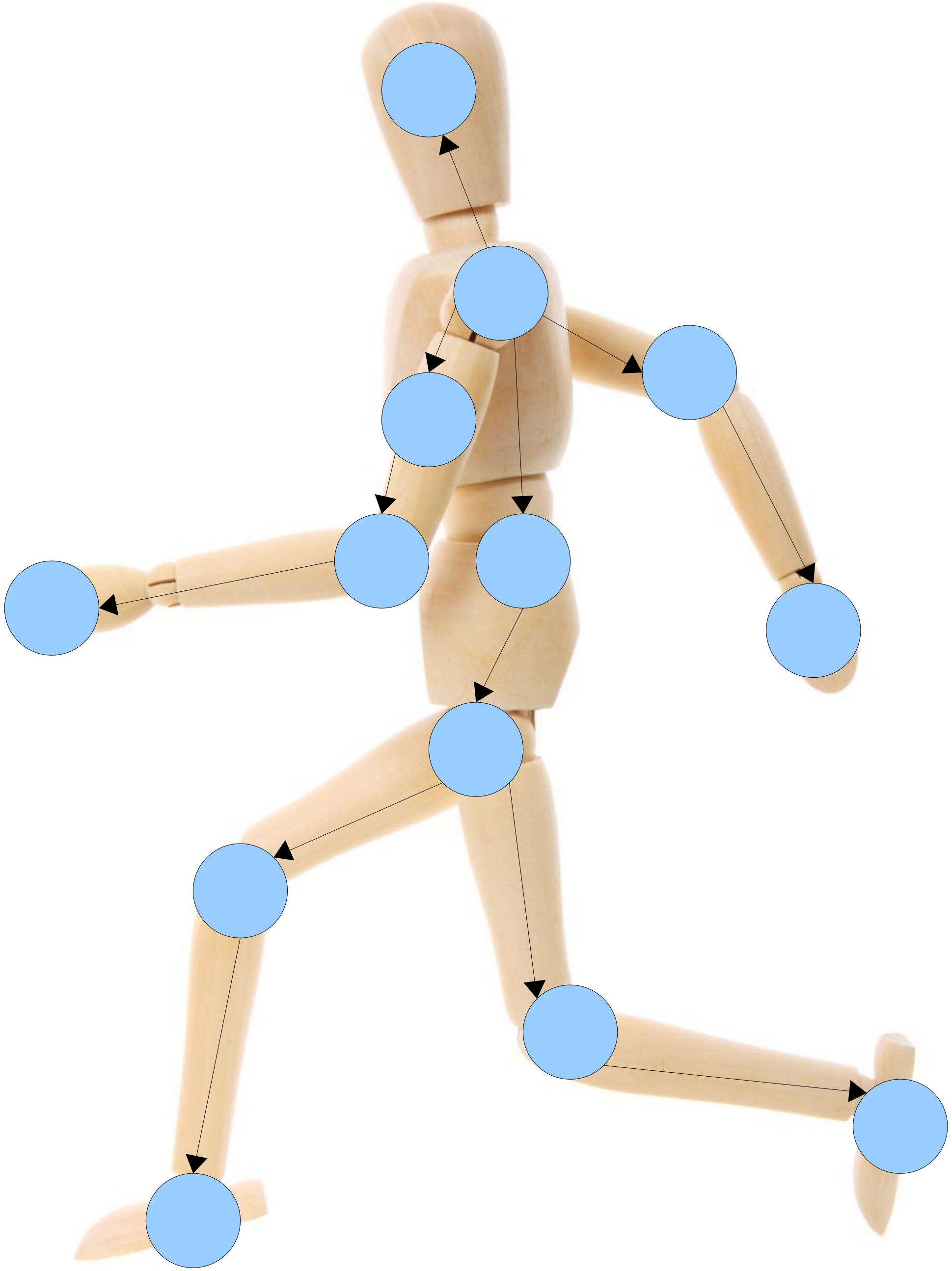
Your alternate explanation has led me to think more critically about this issue and understand why …

Thank you for taking the time to really investigate the data, rather than simply repeating a simple explanation for how …

The data that you are sharing here are indeed taken from that agency, but it’s still a preliminary report. Does that change …

You’re right to identify this. Historically, the initial reports have not changed substantially when the …

http://www.publicdomainpictures.net/pictures/10000/velka/1-1265899974oKJ9.jpg



http://www.publicdomainpictures.net/pictures/10000/velka/1-1265899974oKJ9.jpg

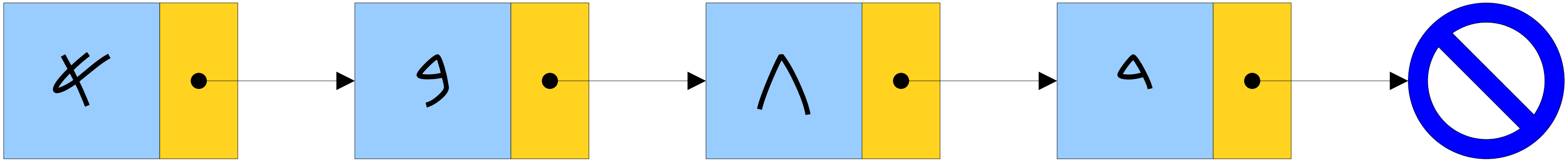
Give another example of a multiway tree.

Answer at
***https://cs106b.stanford.edu/pollev***

# Representing Multiway Trees

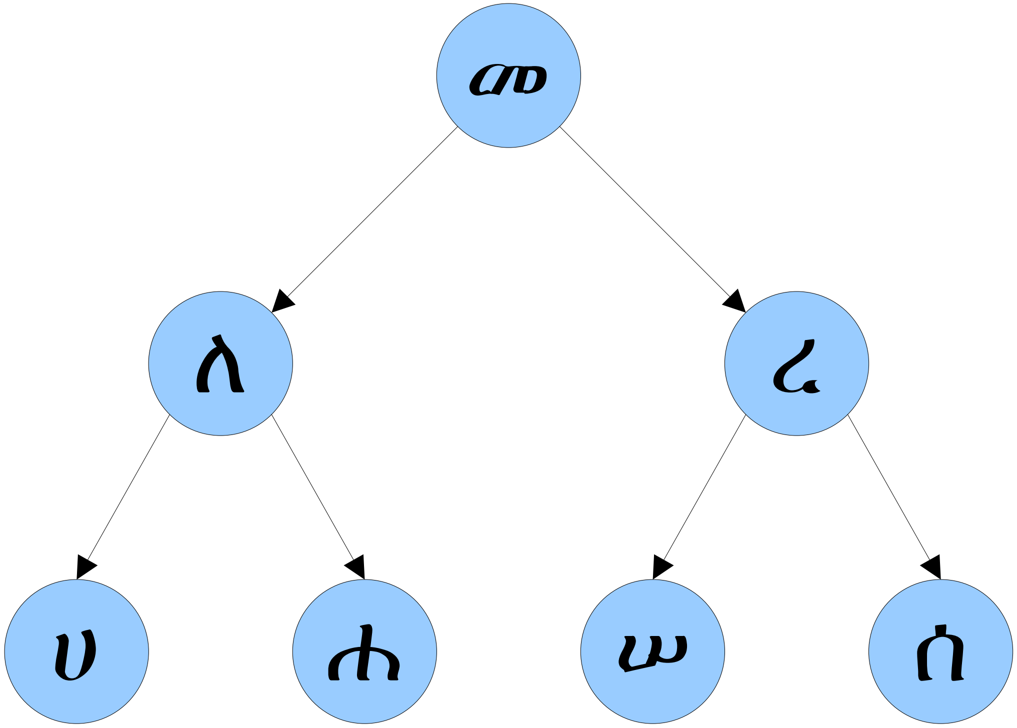
# Representing Multiway Trees

```
struct Cell {
    string data;
    Cell* next;
};
```



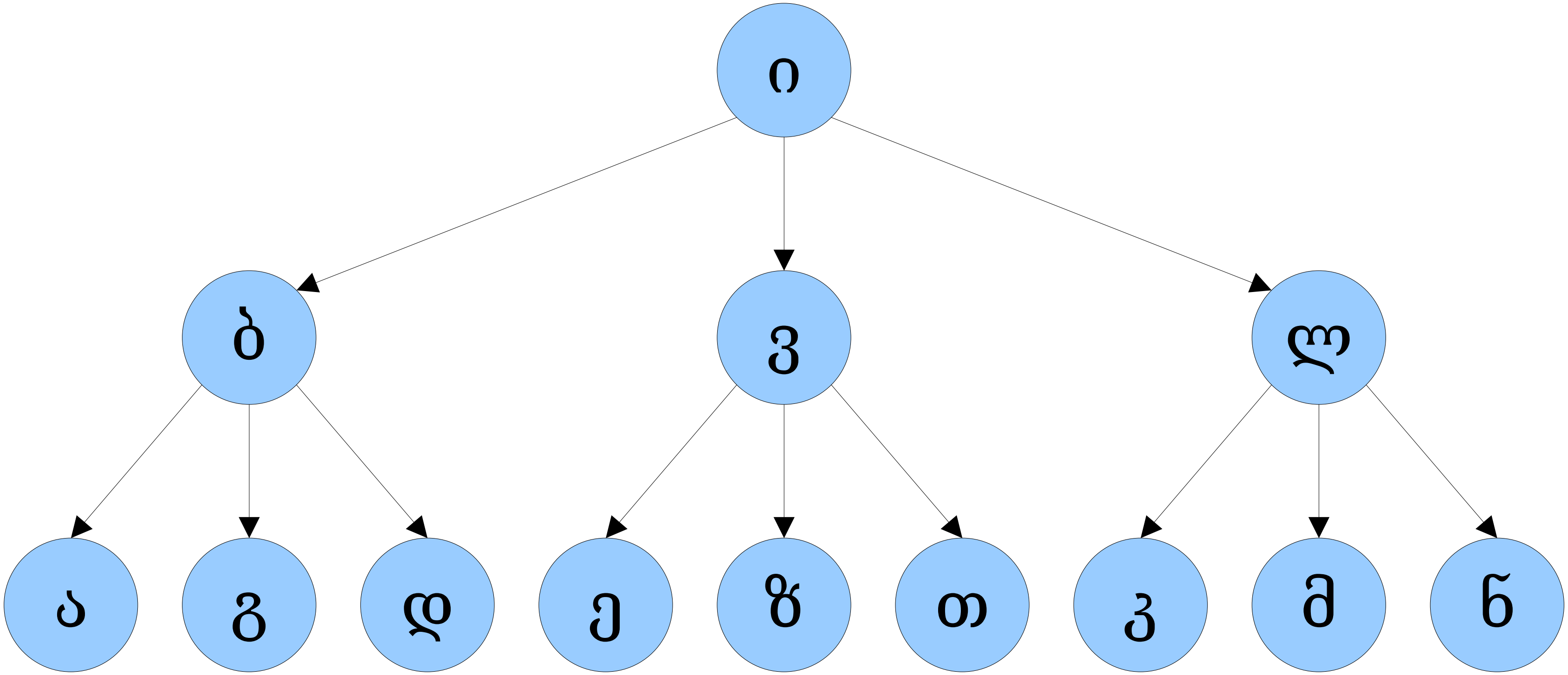
# Representing Multiway Trees

```
struct Node {
    string data;
    Node* left;
    Node* right;
};
```



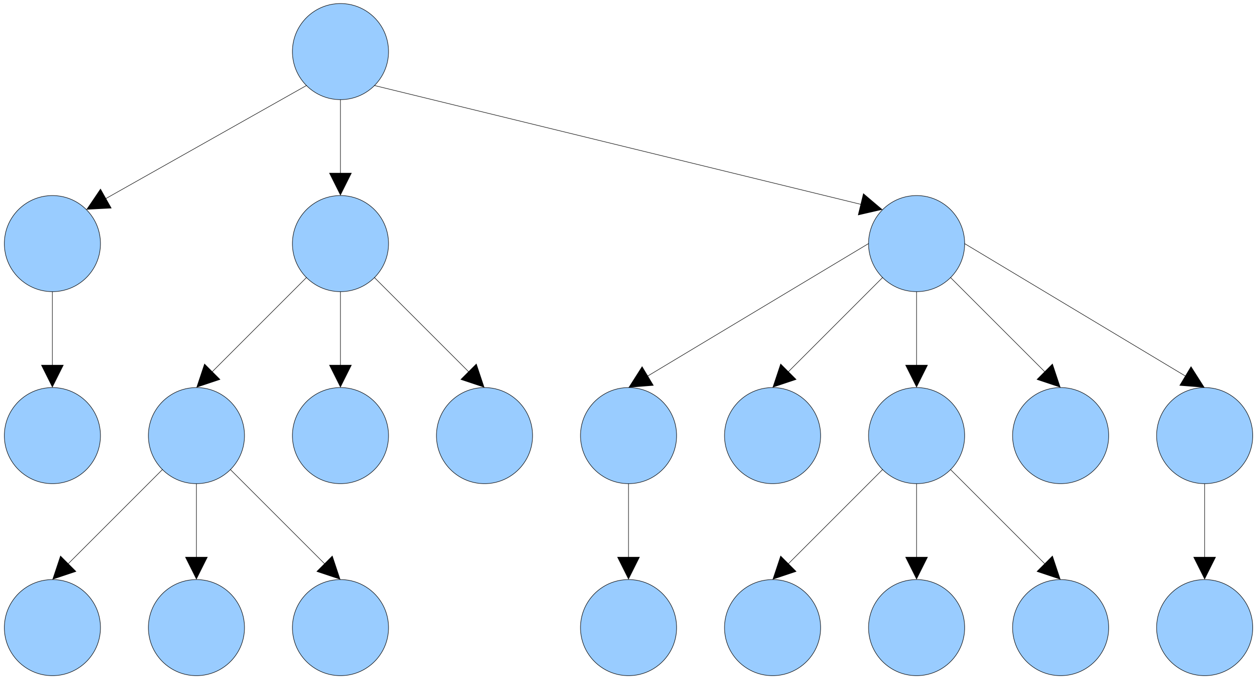
# Representing Multiway Trees

```
struct Node {
    string data;
    Node* left;
    Node* down;
    Node* right;
};
```



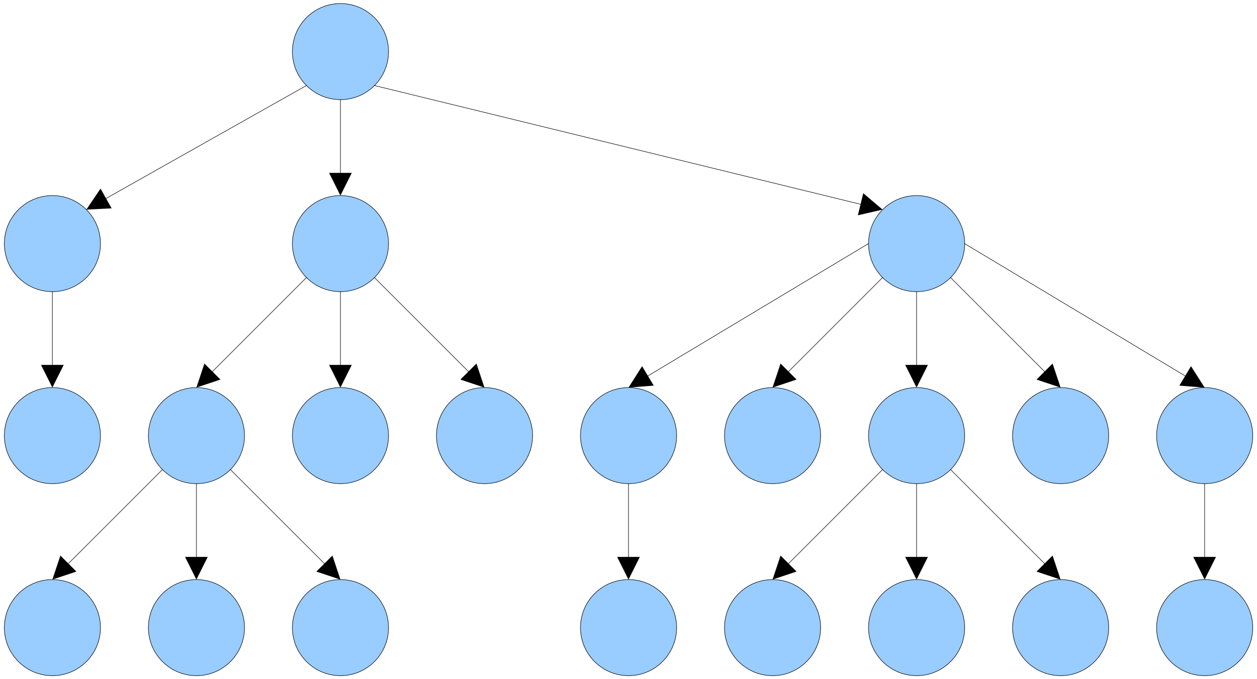
# Representing Multiway Trees

```
struct Node {
    string data;
    Vector<Node*> children;
};
```



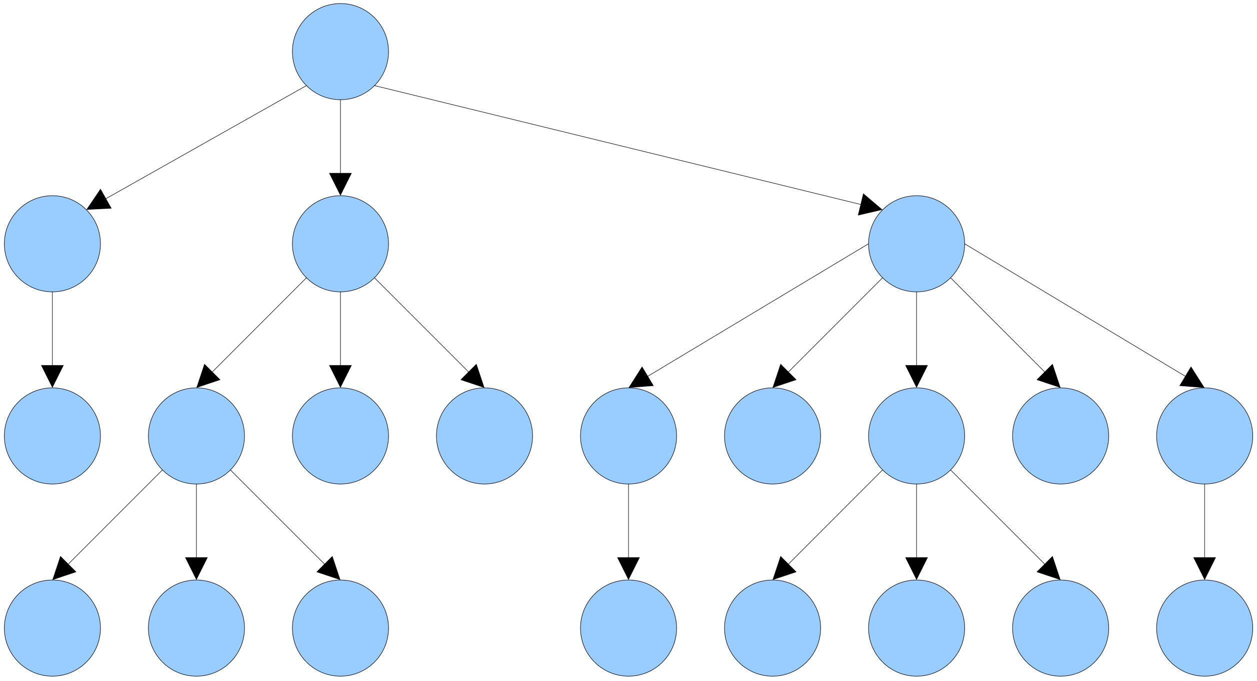
# Representing Multiway Trees

```
struct Node {
    string data;
    Set<Node*> children;
};
```

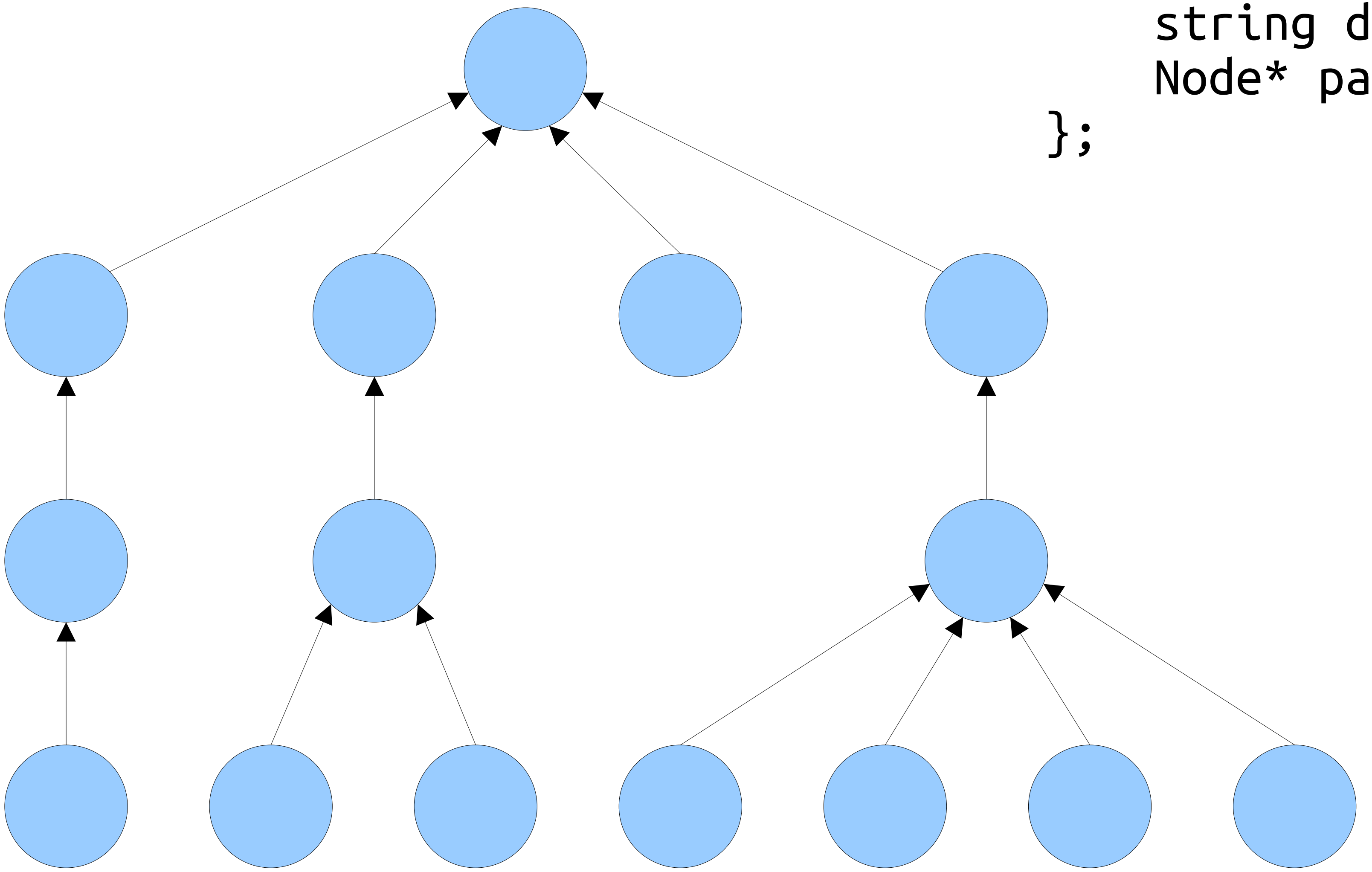


# Representing Multiway Trees

```
struct Node {
    string data;
    Node** children;
    int    numChildren;
};
```

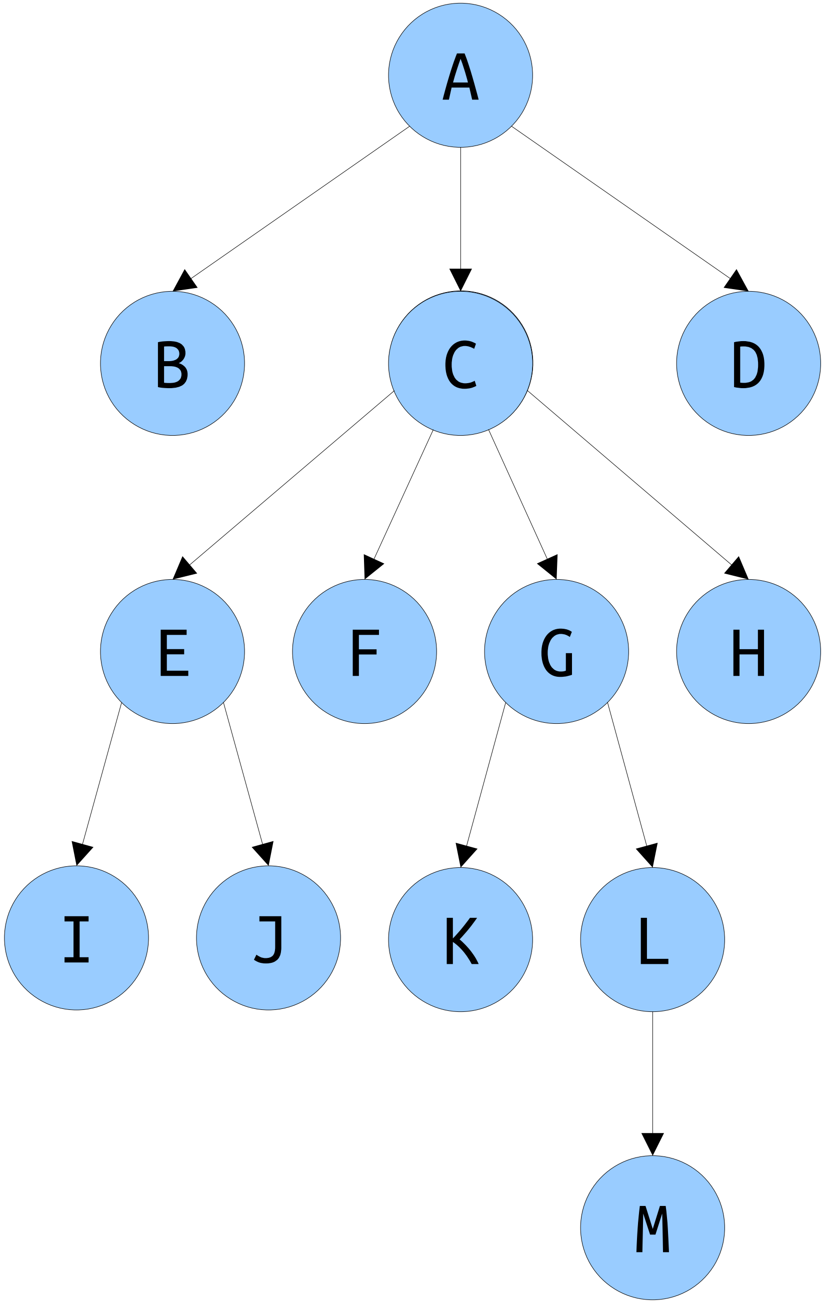


# An Alternative Approach



```
struct Node {
    string data;
    Node* parent;
};
```

# Working with Multiway Trees



```
A {
    B
    C {
        E {
            I
            J
        }
        F
        G {
            K
            L {
                M
            }
        }
        H
    }
    D
}
```

# Working with Real-World Data Sets

# Getting Data

- Many organizations (governments, nonprofits, companies, etc.) release huge quantities of data to the public.
- You have enough coding background at this point to write programs that manipulate, analyze, and visualize these data sets.
  - Often times, you'll need to write more code to get data into your program than you will write to analyze it!
- There are multiple different data file formats out there, of which a few are used frequently when data are released to the public.

# JSON

- ***JSON*** (JavaScript Object Notation) is a popular format for exporting structured data.
- You can find all sorts of JSON data sets out there. For example…
  - The US Geological Survey publishes information on ***all recent earthquakes*** via JSON.
  - The National Park Service lets you query for ***information*** about parks, activities, closures, etc. and get the result in JSON.
- Let’s briefly explore how JSON works.

```
["Hi Mom!", 2.71828, 137, [true, false, true]]
```

JSON can store arrays of values of mixed types.

```
{ "bear":        ["grizzly", "polar", "black", "sun"],
  "cat":         ["bob", "house", "savannah"],
  "hummingbird": ["bee", "topaz", "Anna's"] }
```

JSON can encode “objects,” akin to our `Map` type.

# JSON in C++

- We’ve provided a JSON type that stores JSON data.
- You can create a JSON object from a string or a stream:

```
JSON str  = JSON::parse("[1, 2, 3]");
JSON file = JSON::parse(input);
```

- You can check the type of JSON data and get a view of that data as that type:

```
if (json.type() == JSON::Type::NUMBER) {
    cout << json.asInteger() << endl;
}
```

- You can look up elements by index (arrays) or key (objects):

```
JSON first = json[0];
JSON cute  = json["quokka"];
```

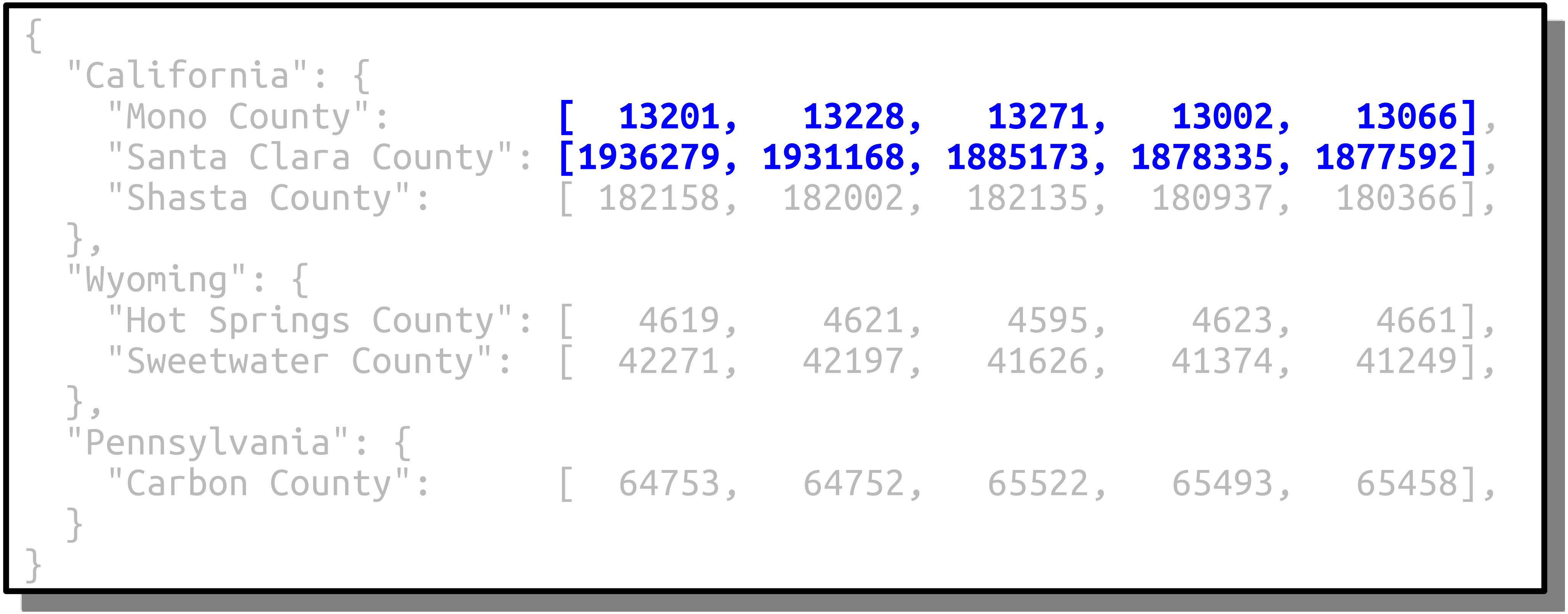
# Our Data Set

- The US Census Bureau ***provides*** information on the population of every US county over the past five years.
- I’ve downloaded this data and converted it into a JSON file.
- ***Our Goal:*** Determine the population of the US in each of the past five years.
  - (We can do a lot more elaborate processing than this; the goal here is to show off the concept.)
- Let’s see what data we have.

```
{
  "California": {
    "Mono County":        [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":      [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":      [  64753,   64752,   65522,   65493,   65458],
  }
}
```

```
{
  "California": {
    "Mono County":         [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":       [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":       [  64753,   64752,   65522,   65493,   65458],
  }
}
```

$[P_{20}, \dots, P_{24}]$



```
{
  "California": {
    "Mono County":        [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":      [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":      [  64753,   64752,   65522,   65493,   65458],
  }
}
```

$[P_{20}, \dots , P_{24}]$

$[P_{20}, \dots , P_{24}]$

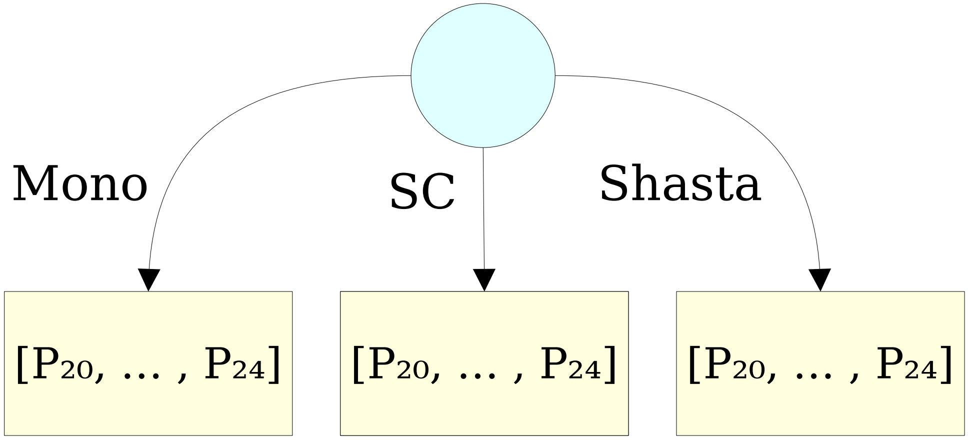
```
{
  "California": {
    "Mono County":        [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":      [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":      [  64753,   64752,   65522,   65493,   65458],
  }
}
```

$[P_{20}, \ldots , P_{24}]$

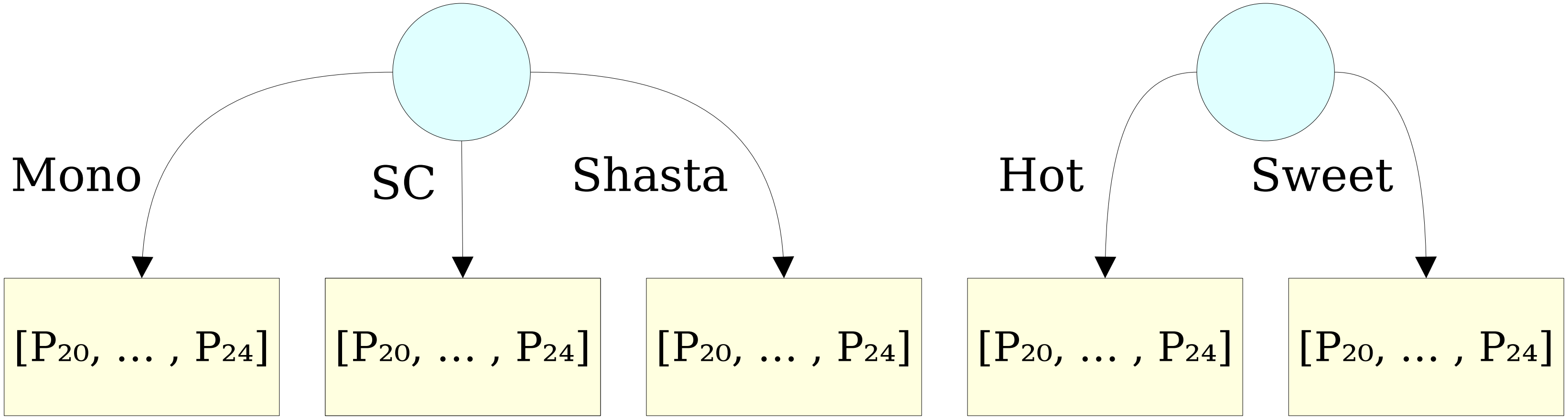
$[P_{20}, \ldots , P_{24}]$

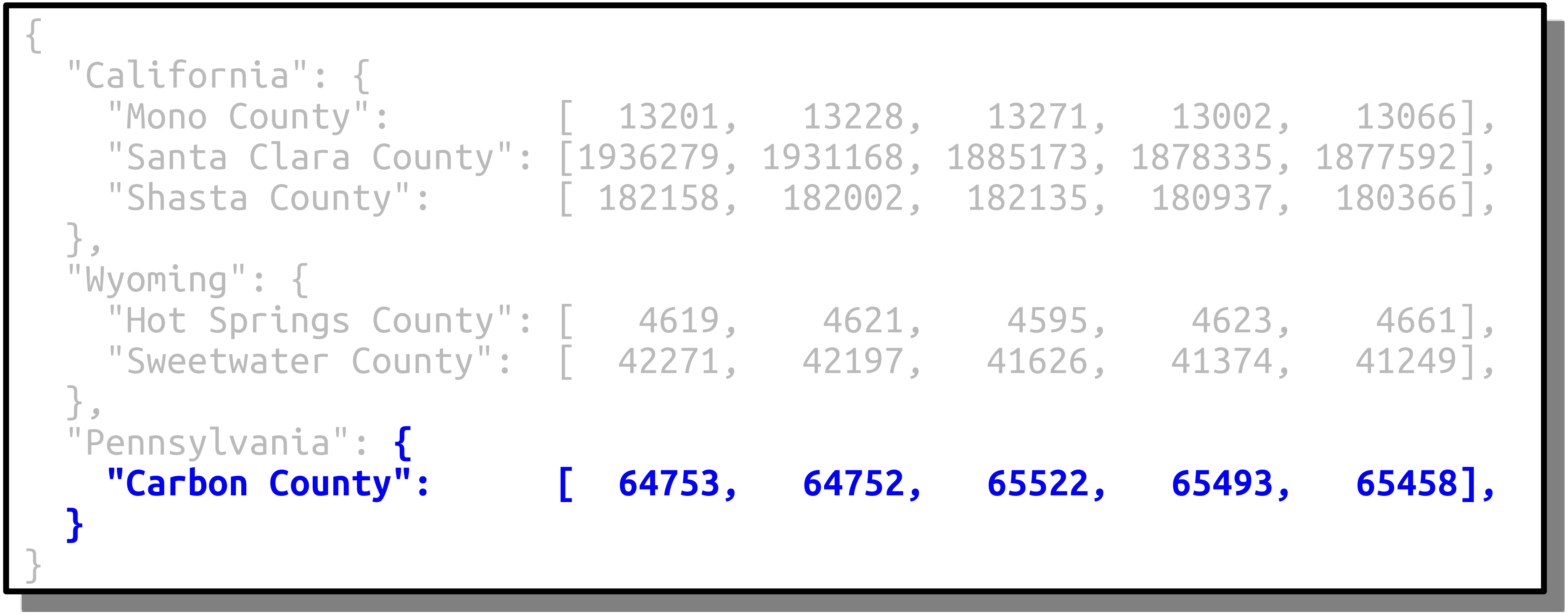
$[P_{20}, \ldots , P_{24}]$

```
{
  "California": {
    "Mono County":          [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":        [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":        [  64753,   64752,   65522,   65493,   65458],
  }
}
```

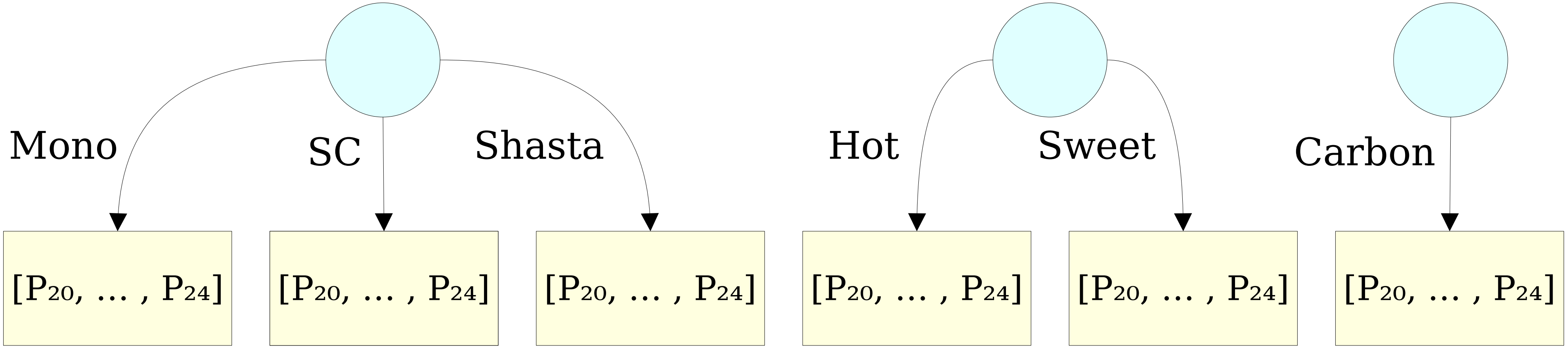


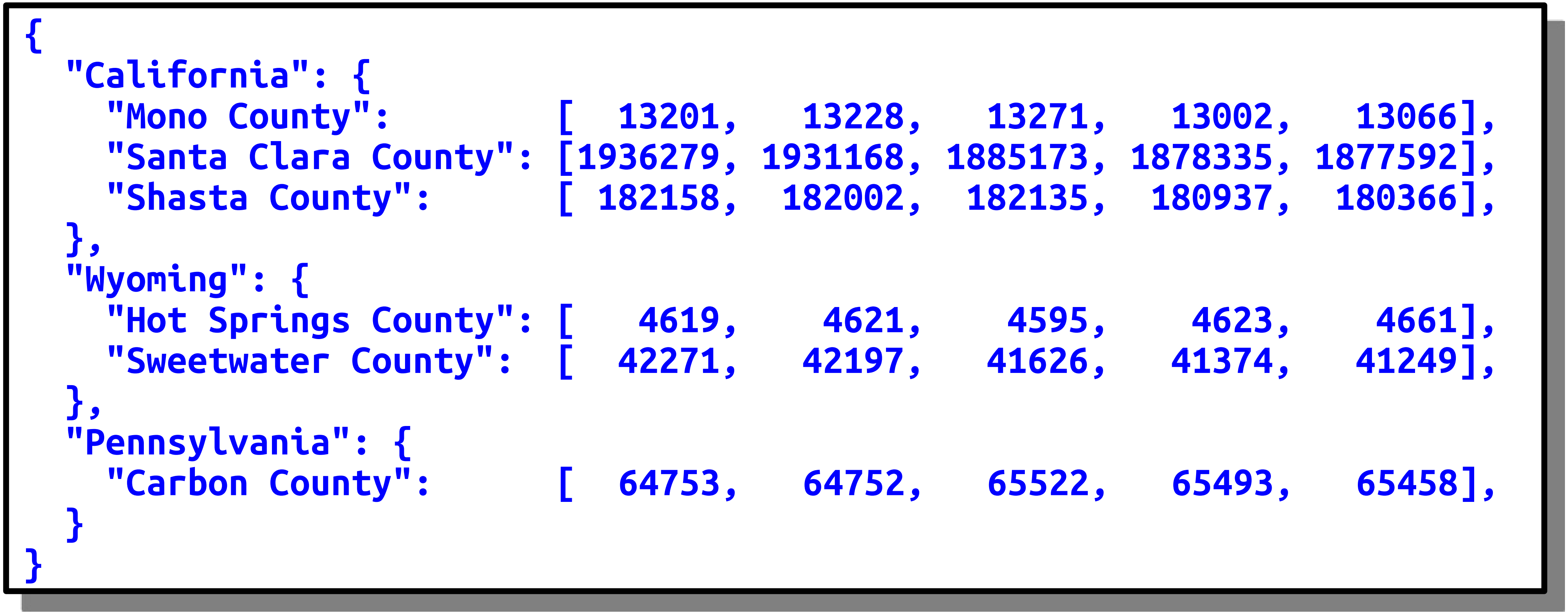
```
{
  "California": {
    "Mono County":        [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":      [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":      [  64753,   64752,   65522,   65493,   65458],
  }
}
```



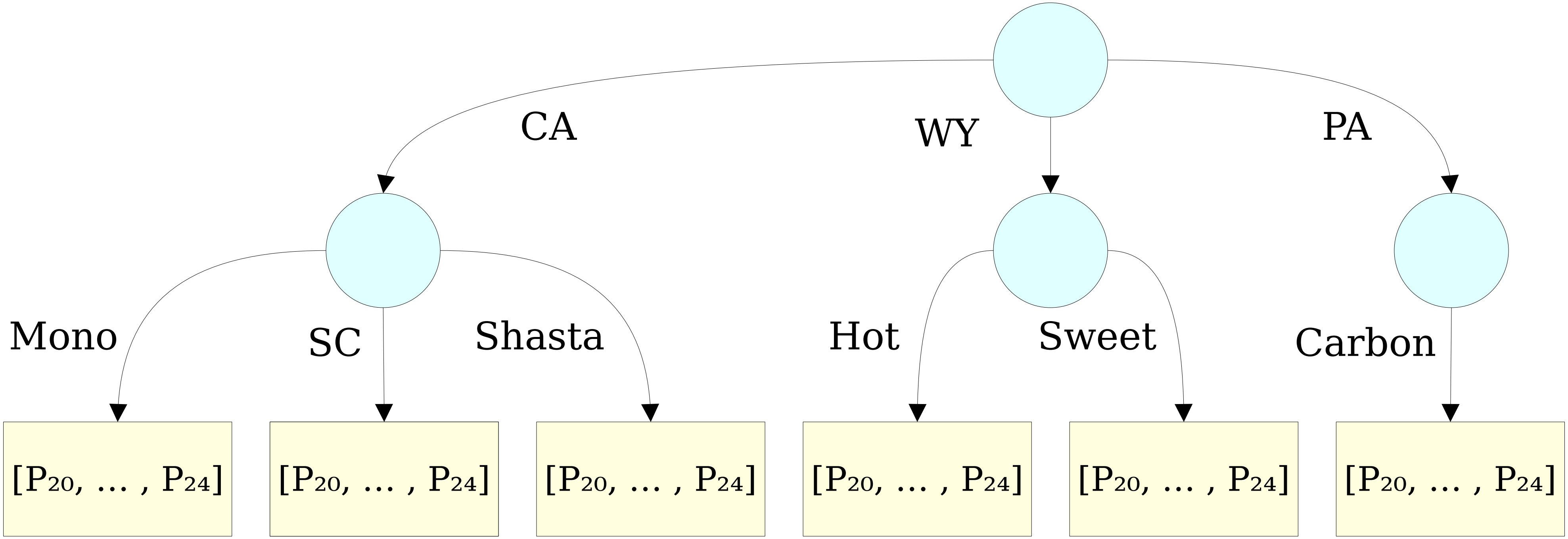


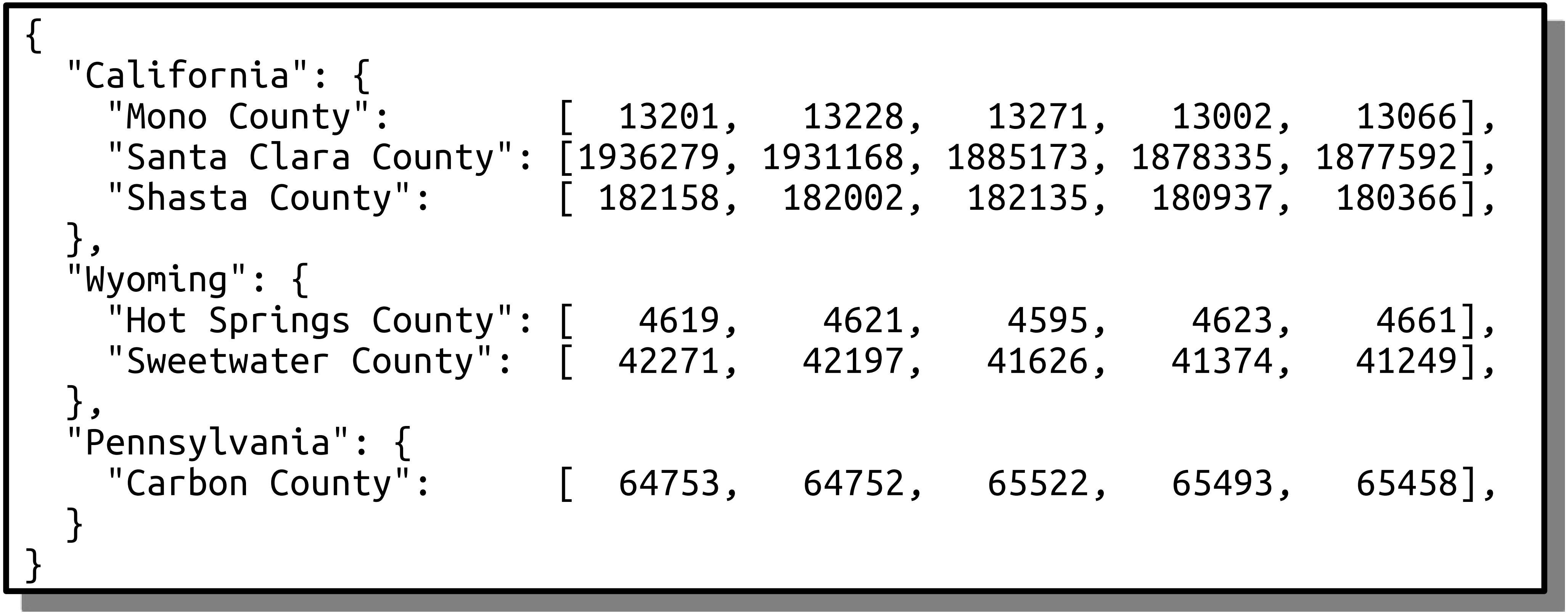
```
{
  "California": {
    "Mono County":         [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":       [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":       [  64753,   64752,   65522,   65493,   65458],
  }
}
```



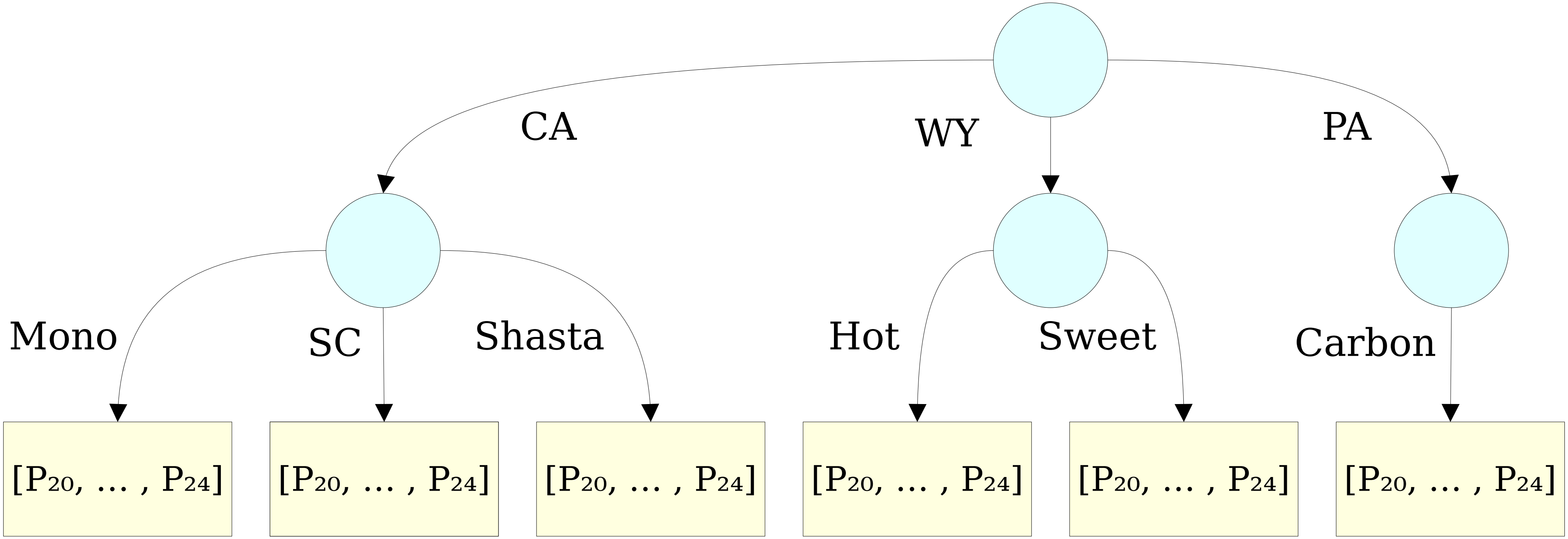


```
{
  "California": {
    "Mono County":          [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":        [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":        [  64753,   64752,   65522,   65493,   65458],
  }
}
```





```
{
  "California": {
    "Mono County":         [  13201,   13228,   13271,   13002,   13066],
    "Santa Clara County": [1936279, 1931168, 1885173, 1878335, 1877592],
    "Shasta County":       [ 182158,  182002,  182135,  180937,  180366],
  },
  "Wyoming": {
    "Hot Springs County": [   4619,    4621,    4595,    4623,    4661],
    "Sweetwater County":  [  42271,   42197,   41626,   41374,   41249],
  },
  "Pennsylvania": {
    "Carbon County":       [  64753,   64752,   65522,   65493,   65458],
  }
}
```



# Your Action Items

- ***Finish Assignment 7***
  - If you’re following our timeline, you’ll have finished the debugger warmup and doubly-linked-list warmups by now, and will have made some progress on Particle Systems.
  - Aim to complete Particle Systems by Friday.
  - Reminder: You can’t use late days on Assignment 8. Be strategic about using them on Assignment 7.