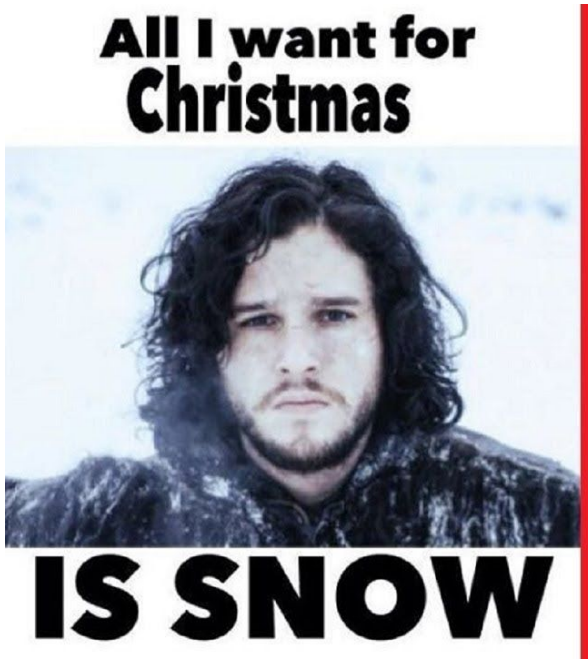
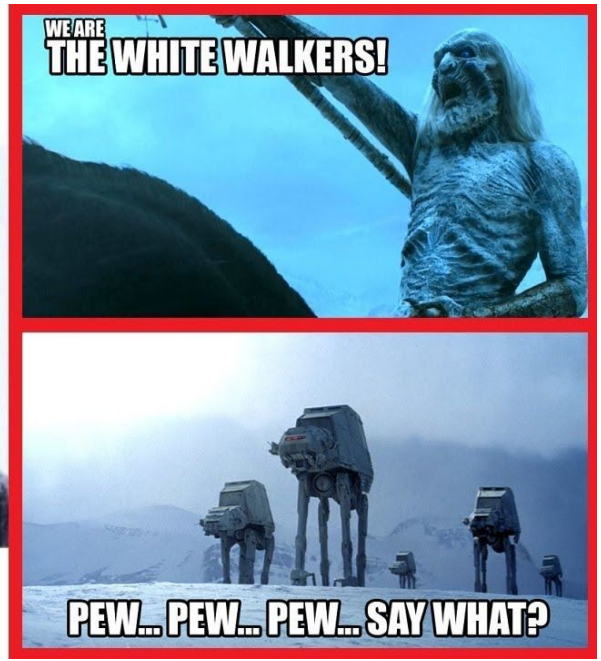
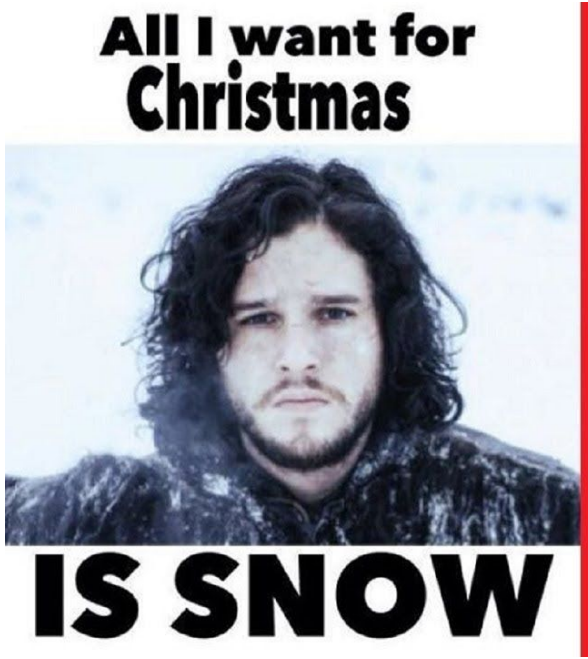
Lecture 30: Managing Large Projects

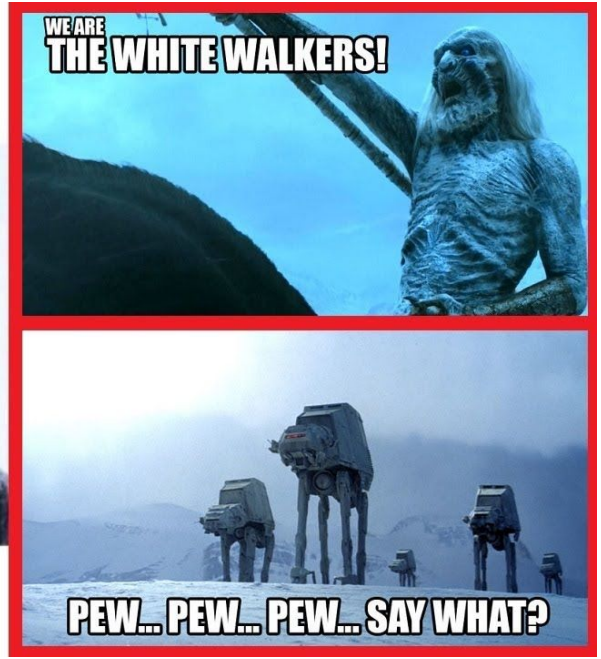




Announcements and reminders

- Project 3 posted
    - By next Wednesday -- TA/CA design meeting
    - … & submit classes and code skeleton
- Project 2 interview grading

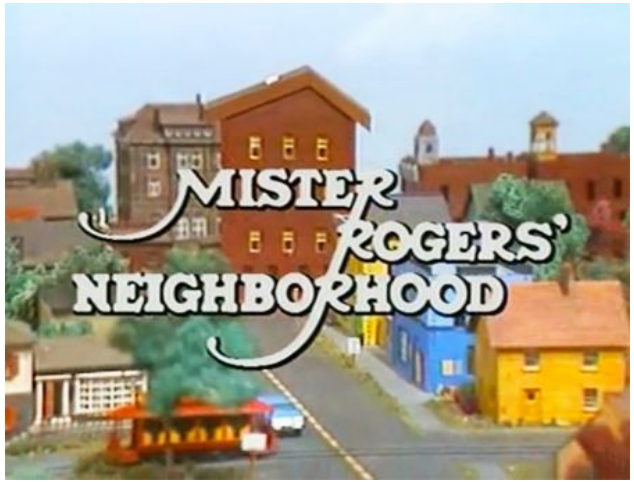




It’s a beautiful day in the neighborhood

Example: Let’s write some code to compute the **median household income** for a town.

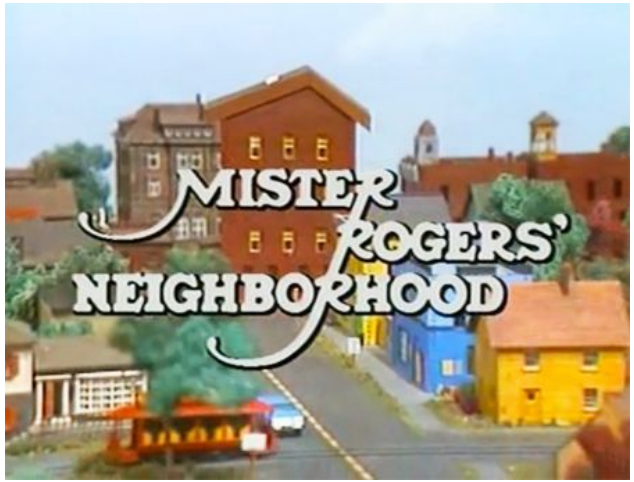
Households each have a **name** and an **income**.



It’s a beautiful day in the neighborhood

Example: The **town** is depicted below and consists of 6 blocks, or **neighborhoods**.

Each neighborhood consists of some number of **households**.



It’s a beautiful day in the neighborhood

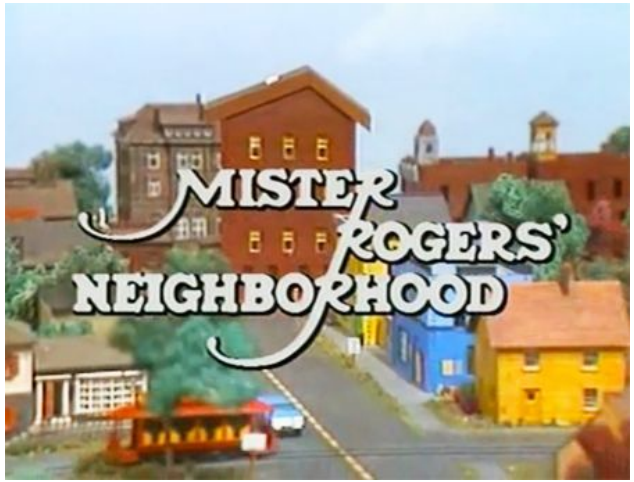
Example: The **town** is depicted below and consists of 6 blocks, or **neighborhoods**.

Each neighborhood consists of some number of **households**.

Highest level: **town**

Middle level: **neighborhood**

Lowest level: **households**

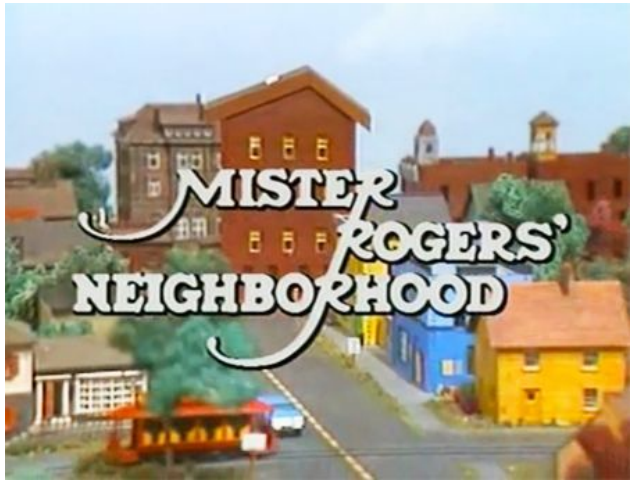


It’s a beautiful day in the neighborhood

Example: The **town** is depicted below and consists of 6 blocks, or **neighborhoods**. Each neighborhood consists of some number of **households**.

What might our **classes** look like here?

- Could represent each **household** individually
    - A **vector** of **households** seems most appropriate. **Why?**



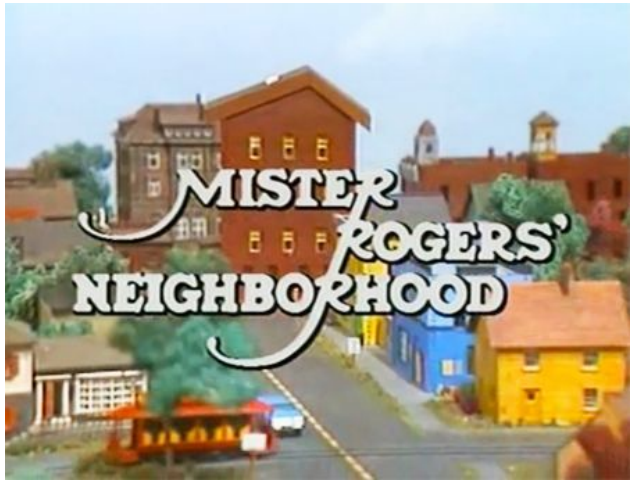
It’s a beautiful day in the neighborhood

Example: The **town** is depicted below and consists of 6 blocks, or **neighborhoods**.
Each neighborhood consists of some number of **households**.

What might our **classes** look like here?

- Could represent each **household** individually
    - A **vector** of **households** seems most appropriate. **Why?**

        → I never told you how many households there are per neighborhood!
        → How’re you going to initialize an array??
- **Problem:** This doesn’t work well if we want to use our program to compute median income for a ***different*** town.

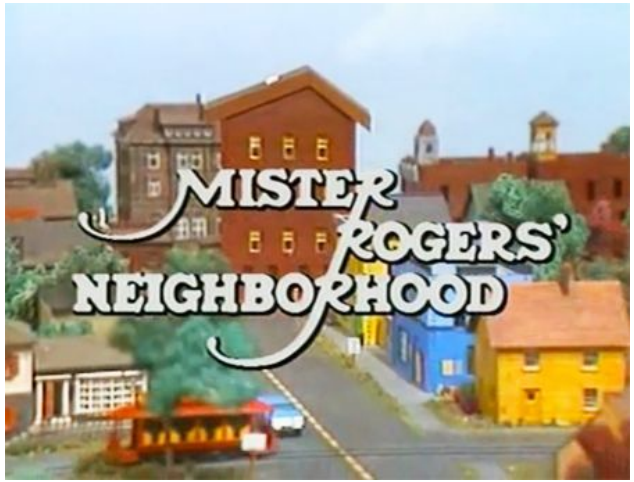


It’s a beautiful day in the neighborhood

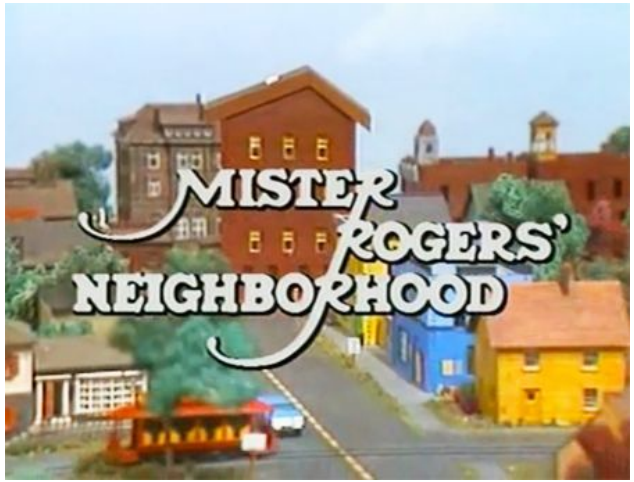
Example: The **town** is depicted below and consists of 6 blocks, or **neighborhoods**.
Each neighborhood consists of some number of **households**.

What might our **classes** look like here?

- Instead, we want to be representing/computing characteristics of a **town**
    - In GOT, a **game** is at the top-level and we are computing different aspects of the game
- **Towns** consist of **neighborhoods**
    - Sounds like a class ⇔ data member relationship!
- But **neighborhoods** consist of **households**
    - Another class ⇔ data member relationship!



It’s a beautiful day in the neighborhood



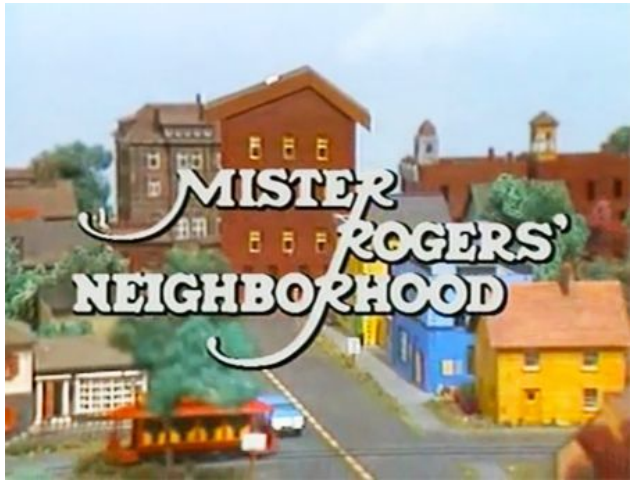
It’s a beautiful day in the neighborhood

```
class Town {
…
private:
    vector<Neighborhood> hoods;
};


class Neighborhood {
…
private:
    vector<Household> houses;
};


class Household {
…
private:
    string name;
    double income;
};
```

If the geographical arrangement of the neighborhoods mattered, what should we do?



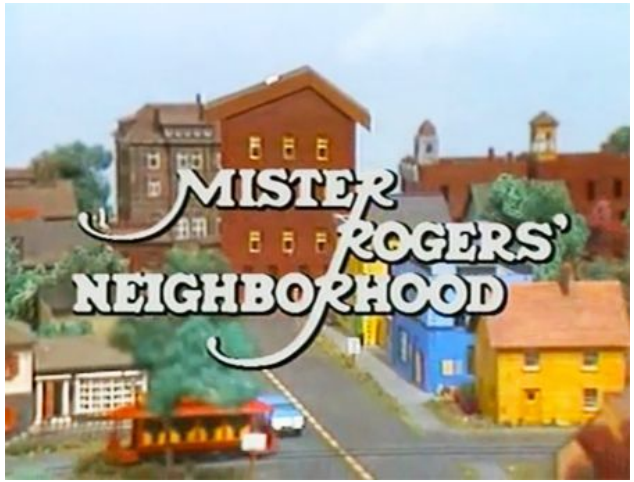
It’s a beautiful day in the neighborhood

```
class Town {
…
private:
    vector<Neighborhood> hoods;
};
```

If the geographical arrangement of the neighborhoods mattered, what should we do? → **Use a 2D array!**

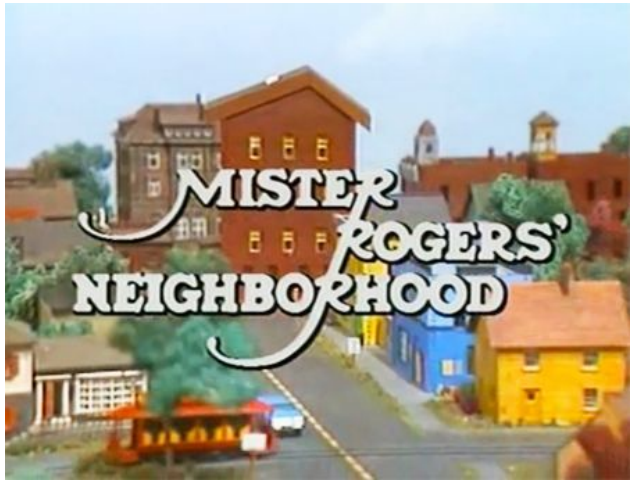
```
class Neighborhood {
…
private:
    vector<Household> houses;
};
```

```
class Household {
…
private:
    string name;
    double income;
};
```



It’s a beautiful day in the neighborhood

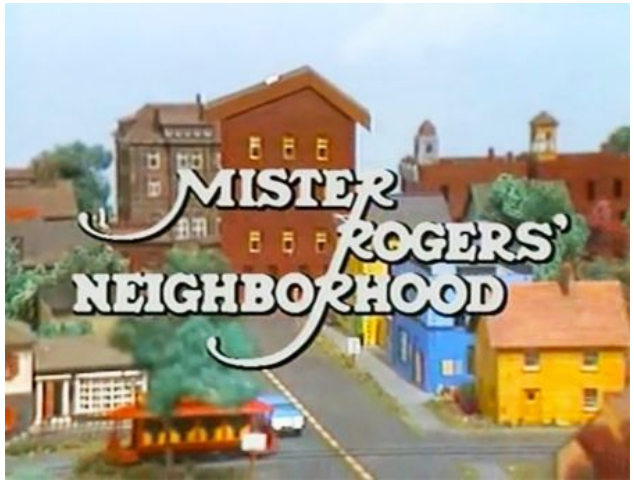
- What’s the key member function for this problem?
- What class is it a member function of?
- What does it need to do?



It’s a beautiful day in the neighborhood

- What’s the key member function for this problem? → **computeMedianIncome()**
- What class is it a member function of? → **Town** (since this is a property of the town)
- What does it need to do?

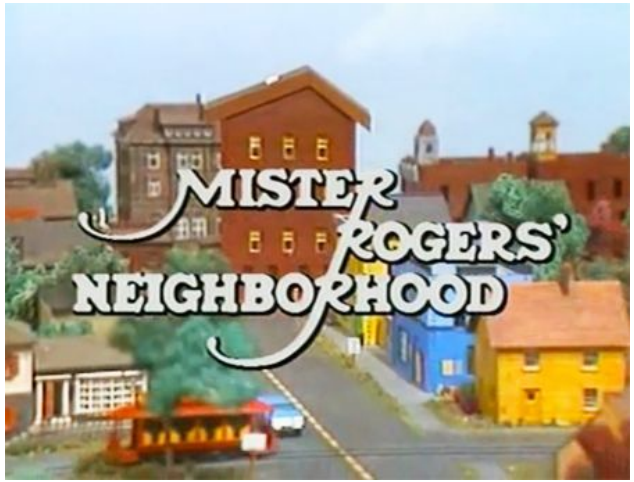
```
class Town {
public:
    ...
    double computeMedianIncome();
    ...
private:
    vector<Neighborhood> hoods;
};
```



It's a beautiful day in the neighborhood

- What's the key member function for this problem? → **computeMedianIncome()**
- What class is it a member function of? → **Town** (since this is a property of the town)
- What does it need to do?
  - → Loop over all the **Households**, in all the **Neighborhoods**, and tally up all of the household incomes.
  - → Compute the **median** of all the incomes

```
class Town {
public:
    ...
    double computeMedianIncome();
    ...
private:
    vector<Neighborhood> hoods;
};
```



It's a beautiful day in the neighborhood

- What's the key member function for this problem? → **computeMedianIncome()**
- What class is it a member function of? → **Town** (since this is a property of the town)
- What does it need to do?

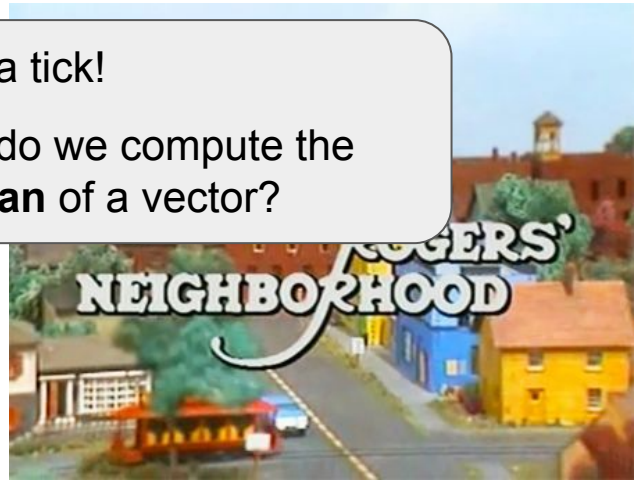
  → Loop over all the **Households**, in all the **Neighborhoods**, and tally up all of the household incomes.

  → Compute the **median** of all the incomes

```
class Town {
public:
    ...
    double computeMedianIncome();
    ...
private:
    vector<Neighborhood> hoods;
};
```

Wait a tick!

How do we compute the **median** of a vector?



Quick detour: how do I median?

The **median** of a set of n data points **X** = [x_0, x_1, …, x_{n-1}] is the **middle-most data point** when the data set is sorted from smallest to largest.

- If n is odd, then there is a single data point right in the middle
- If n is even, then the median is the **mean** of the two middle-most values

Examples:

X = [1, 3, 9, 13, 5] → sort(X) = [1, 3, 5, 9, 13] → median(X) = 5

X = [1, 9, 13, 5] → sort(X) = [1, 5, 9, 13] → median(X) = (5+9)/2 = 7

Quick detour: how do I median?

The **median** of a set of n data points **X** = [x_0, x_1, …, x_{n-1}] is the **middle-most data point** when the data set is sorted from smallest to largest.

- If n is odd, then there is a single data point right in the middle
- If n is even, then the median is the **mean** of the two middle-most values

Examples:

X = [1, 3, 9, 13, 5] → sort(X) = [1, 3, 5, 9, 13] → median(X) = 5

X = [1, 9, 13, 5] → sort(X) = [1, 5, 9, 13] → median(X) = (5+9)/2 = 7

→ So, we need a helper function to compute the **median** of a vector or array, X

→ But ***that*** is going to also require ***another*** helper function to **sort** a vector or array

It's a beautiful day in the neighborhood

- What's the key member function for this problem? → **computeMedianIncome()**
- What class is it a member function of? → **Town** (since this is a property of the town)
- What does it need to do?

  → Loop over all the **Households**, in all the **Neighborhoods**, and tally up all of the household incomes.

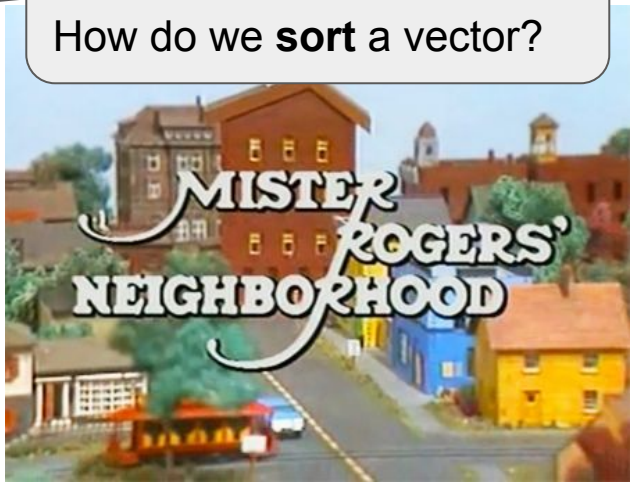
  → Compute the **median** of all the incomes

  → **Sort** the list of median incomes

```
class Town {
public:
    ...
    double computeMedianIncome();
    ...
private:
    vector<Neighborhood> hoods;
};
```

Hold up!

How do we **sort** a vector?



Yet another detour: how do I sort?

Input: An array or vector of n data points **X =** $[x_0, x_1, \ldots, x_{n-1}]$ in **arbitrary order**

Output: The **sorted** version of **X**, in increasing order

There are **LOTS** of ways to do this!

- **Fastest sorting algorithms:** quick sort and merge sort

We'll start smaller: **selection sort** (Special Topic 6.2 in the textbook)

Selection sort (Special Topic 6.2)

Input: **X** = [13, 3, 9, 5, 1]

Output: The **sorted** version of **X**, in increasing order: [1, 3, 5, 9, 13]

Step 1: Find the smallest element out of X[0 - end].
Swap X[0] and smallest element.

Step 2: Find the smallest element out of X[1 - end].
Swap X[1] and smallest element.

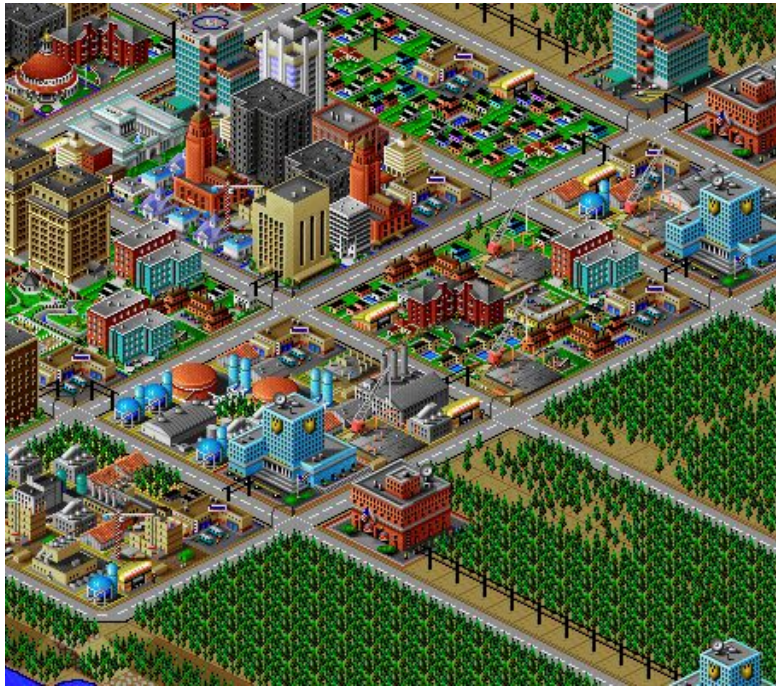
Step 3: Find the smallest element out of X[2 - end].
Swap X[2] and smallest element.

And so on…

Let’s think about what we just did (or are in the middle of doing)

We looked at **what to do when we’re handed a big project**

- Identify what are the key **structures**
- … and how those structures **relate** to one another
- Identify what are the key **functions**
- … and how these functions are related to our structures



Important take-away: decomposing big problems into smaller ones

- What's the key member function for this problem? → **computeMedianIncome()**
- What class is it a member function of? → **Town** (since this is a property of the town)
- What does it need to do?

  → Loop over all the **Households**, in all the **Neighborhoods**, and tally up all of the household incomes.

  → Compute the **median** of all the incomes

  → **Sort** the list of median incomes

```
class Town {
public:
    ...
    double computeMedianIncome();
    ...
private:
    vector<Neighborhood> hoods;
};
```