A Systematic Approach to Find the Best City to Live Based on Climate Change Dataset

Steven Phillips, Dr. Aravind Mohan
Department of Computer Science, McMurry University, Abilene, TX 79697

Abstract

In today's market, many factors influence the choice of finding a new place to live in the United States of America. The most common factors that come to mind first are Money, Time & Accessibility. These factors are the first key points brought up when looking for a house in a new state or new city. However, a factor that is commonly left out of that conversation is climate! Among the factors that were listed; money, time, accessibility, and climate, the climate is the most unpredictable factor when looking for a place to live, specifically the weather. In recent years, Climate Change has been an increasing concern in the United States of America as there has been a rise in natural disasters and high temperatures. As we continue to tackle the problems that climate change brings, this creates another factor to consider when searching for a new home in America. This factor wasn't considered before, but as our climate continues to change, it should be a definitive factor when choosing a place to live. While predicting current weather can be a struggle, some resources can be used to help relieve those issues. With the dataset provided by NCAA.gov, we specified the rising and falling temperatures of cities across the United States. Through our implementation using C, a structured programming language, and experimentation, we demonstrated the accuracy and efficiency of our tool to help those looking for a new place to live.

Introduction

Finding a place to live in the United States of America has become a challenging task for young millennials as factors towards finding the perfect home have become increasingly difficult. A factor that has become a huge problem in recent years is climate change. According to 'UnitedNations.org', Climate Change refers to long-term temperature shifts and weather patterns. Due to human activity as of the late 1800s, the weather has become more unpredictable with each passing month. The burning of fossil fuels specifically has caused rising temperatures in America over the course of the past ten years. According to a research project by the 'University of California', the burning of fossil fuels is the primary cause of current climate change, altering the Earth's ecosystem and causing environmental health problems. As we continue to burn more natural resources, we continue to change the climate and weather around us. The first solution that would come to mind would be to stop using those natural resources so frequently. However, as technology continues to arise in the market, so are the natural resources being burned. Climate factors into finding a place to live. Each citizen has specific needs and necessities that the weather of the city and state that they live in can either benefit or hinder. Examples such as farmers needing a place to live that will have sunlight to grow their crops over the course of the year. As the weather has become more unpredictable over the course of the last ten years, reliable information and resources when it comes to finding cities that meet specific needs have become harder to find. In this research, we will utilize the data set from the National Centers for Environmental Information and the C programming language to construct a comprehensive data model following a structured programming approach. By tracking the weather for each city in America, our methodology involves categorizing the maximum and minimum average temperature of all cities based on the seasons of the year. Using our algorithm, Best City Finder (BCF), We display the hottest and coldest cities in America based on seasons. Additionally, our experiments demonstrate that the algorithm finds the best place to live efficiently and accurately.

Methodology

The Best City Finder (BCF) algorithm, as shown in Table 5, is designed to analyze a weather dataset, D, containing the minimum and maximum average temperatures of all the cities in America based on the user's input choice, C. Firstly, the algorithm extracts the data from the dataset and places the temperature extracted, recording, in one of the four seasons based on what season the temperature is being pulled from. When prompted, if the user selects "Hottest", the algorithm will sort the temperatures for each season in descending order and display the top five hottest temperatures with their corresponding locations (city, state) for all four seasons, S1*, S2*, S3*, and S4*. For example, in Table-1, Table-2, Table-3, Table-4, we showed the sample results for the Winter, Spring, Summer, and Fall seasons. Let us suppose that the user choice is hottest, then the top 5 hottest locations are displayed to the user with Key West, FL listed as the hottest location. In contrast, let us suppose that the user choice is coldest, the top 5 coldest locations are displayed to the user with Barrow, AL listed as the coldest location.

Table 1: Sample Output for the Winter Season

City, State	Hottest Temp.	City, State	Coldest Temp.
Key West, FL	74.90	Barrow, AK	-9.70
Hilo, HI	73.40	Inter. Falls, MN	7.80
Brownsville, TX	67.70	Mt. Washington, NH	10.60
Death Valley, CA	66.20	Lowell, MA	29.40
Phoenix, AZ	64.30	Ely, NV	29.90

Table 2: Sample Output for the Spring Season

City, State	Hottest Temp.	City, State	Coldest Temp.
Bonita Springs, FL	86.90	Barrow, AK	14.70
Brownsville, TX	85.80	Inter. Falls, MN	22.80
Honolulu, HI	81.90	Mt. Washington, NH	30.50
Yuma, AZ	78.70	Ely, NV	36.50
Wilmington, NC	75.70	Rochester, NY	40.20

Table 3: Sample Output for the Summer Season

City, State	Hottest Temp.	City, State	Coldest Temp.
Lihue, HI	91.10	Nome, AK	21.30
Brownsville, TX	90.30	Inter. Falls, MN	36.70
Miami, FL	90.00	Elkins, WV	40.40
Monroe, LA	83.30	Spokane, WA	40.50
Yuma, AZ	81.70	Rochester, NY	41.60

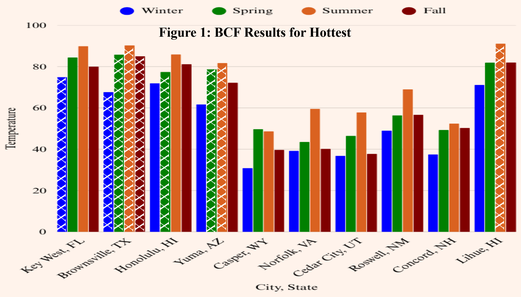
Table 4: Sample Output for the Fall Season

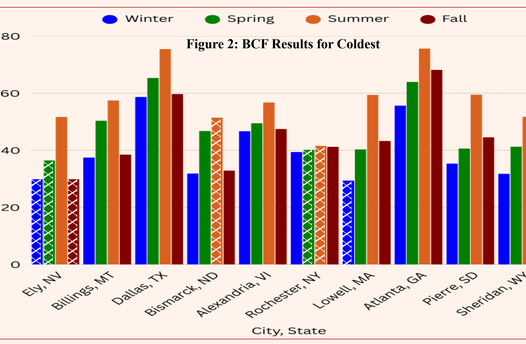
City, State	Hottest Temp.	City, State	Coldest Temp.
Miami, FL	85.60	Fairbanks, AK	15.10
Monroe, LA	84.30	Ely, NV	29.90
Hilo, HI	82.40	Sheridan, WY	32.80
Reno, NV	82.30	Bismarck, ND	32.90
Brownsville, TX	78.40	Kalispell, MT	32.90

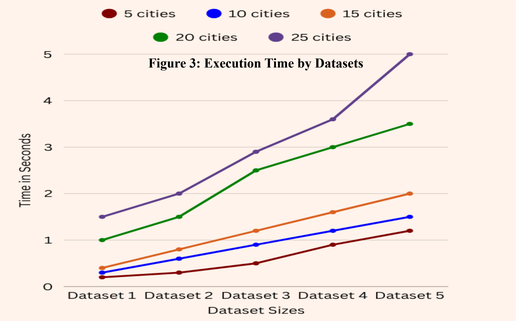
Table 5: An Overview of the BCF Algorithm

```
Input: Choice C, Dataset, D
Output: Seasons, S1*, S2*, S3*, S4*
S1.Init(), S2.Init(), S3.Init(), S4.Init()
While (D.Item == NULL)
   recording = D.nextItem()
   if (12/21 >= recording.date <= 03/20)
      return S1.add(recording)
   else if (03/21 >= recording.date <= 06/20)
      return S2.add(recording)
   else if (06/21 >= recording.date <= 09/20)
      return S3.add(recording)
   else if (09/21 >= recording.date <= 12/20)
      return S4.add(recording)
End While
Switch (C):
   Case Hottest:
      S1.Sort(Descending by temp)
      S2.Sort(Descending by temp)
      S3.Sort(Descending by temp)
      S4.Sort(Descending by temp)
      For (int i → 0 to 5; i++)
         S1*.add(S1.get(i).get(loc, temp))
         S2*.add(S2.get(i).get(loc, temp))
         S3*.add(S3.get(i).get(loc, temp))
         S4*.add(S4.get(i).get(loc, temp))
      End For
   Case Coldest:
      S1.Sort(Ascending by temp)
      S2.Sort(Ascending by temp)
      S3.Sort(Ascending by temp)
      S4.Sort(Ascending by temp)
      For (int i → 0 to 5; i++)
         S1*.add(S1.get(i).get(loc, temp))
         S2*.add(S2.get(i).get(loc, temp))
         S3*.add(S3.get(i).get(loc, temp))
         S4*.add(S4.get(i).get(loc, temp))
      End For
End Switch
return S1*, S2*, S3*, S4*
```

Experiments







We randomly picked locations from the top 5 candidates and locations not from the top 5 candidates to generate our figures 1 and 2. In Figure 1, we demonstrated the accuracy of our tool by showing the locations that were ranked in the top 5 hottest cities in all four seasons. In Figure 2, we demonstrated the accuracy of our tool by showing the locations that were ranked in the top 5 coldest cities in all four seasons. In Figure 3, we demonstrated the efficiency of our tool by increasing the size of our dataset and calculating how much time the tool took to run when ranking the top 5, top 10, top 15, top 20, and top 25 cities. As the dataset increased, so did the time.

Conclusion

The BCF algorithm perfectly executes an efficient and accurate data model that allows the user to receive the information needed to choose their home. This avoids the hassle of looking around the web for certain weather-related information that is required to be considered. Our implementation of the algorithm using the structured programming approach had helped us to accurately and efficiently model the data and find the best city to live in.