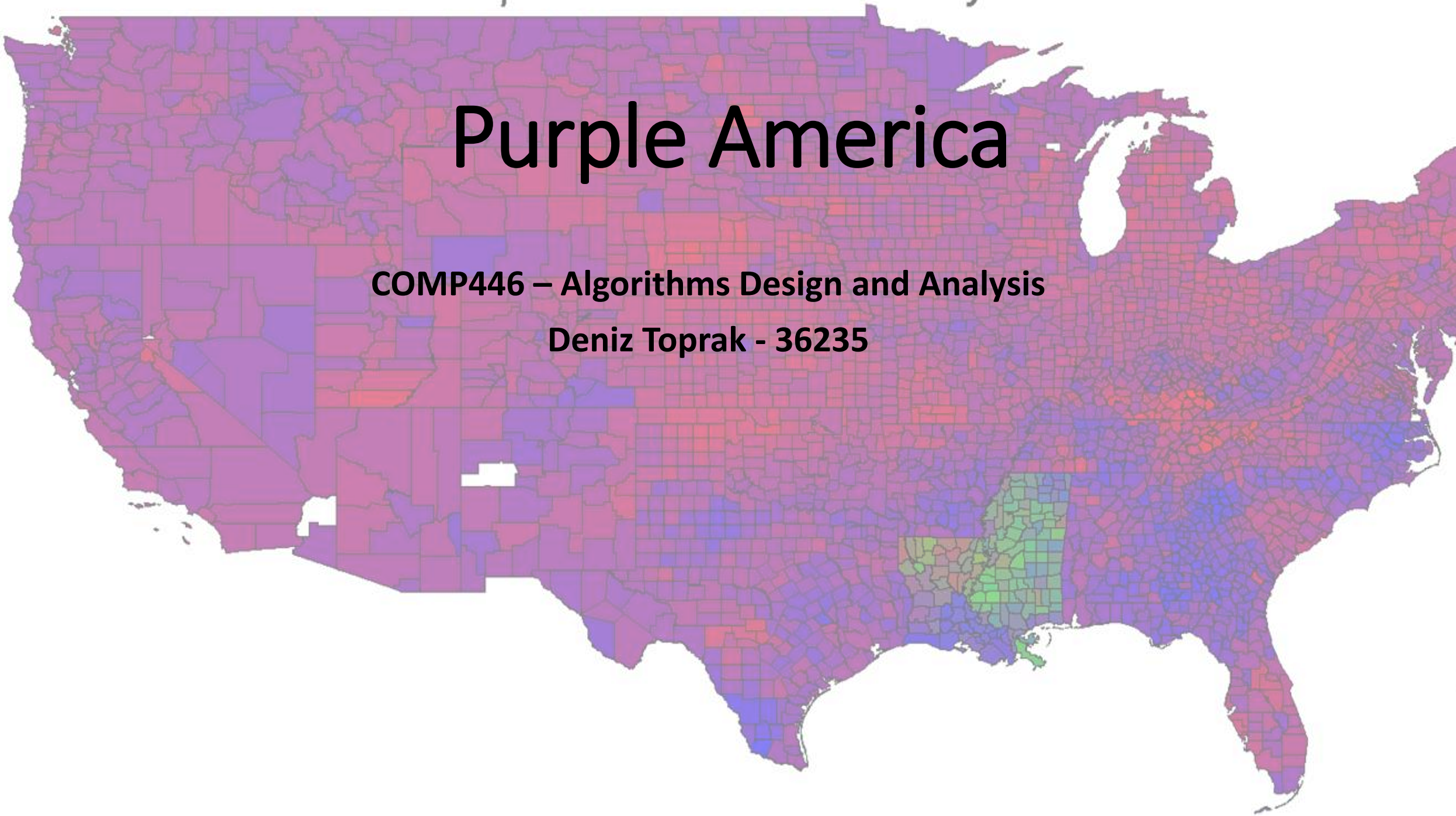


# Purple America

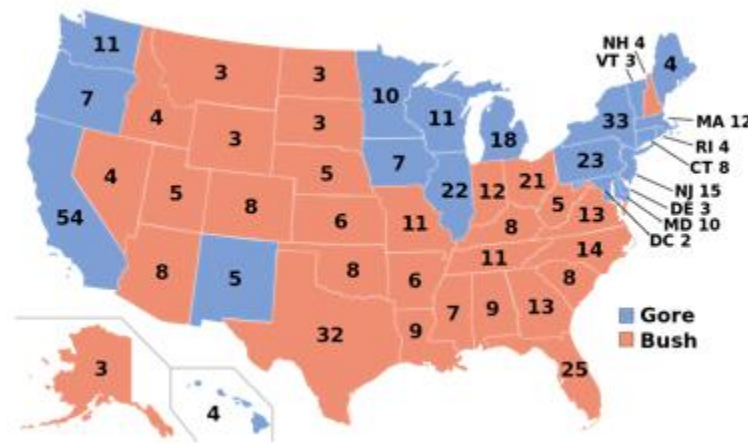
**COMP446 – Algorithms Design and Analysis**

**Deniz Toprak - 36235**



# Historical Information

During coverage of the 2000 presidential election, Tim Russert coined the political terms red states and blue states to refer to states that predominantly vote for the Republican presidential candidate (red) or the Democratic presidential candidate (blue). The news media use red-state blue-state maps, such as the one below, to display election results.

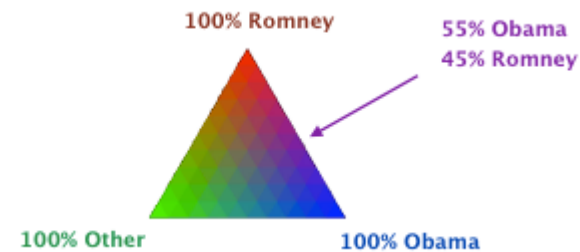


Tim Russert  
Meet the Press

# Purple America

Write a program Purple.java that takes two command-line arguments (the name of the map and the year of the election) and produces a Purple-America map. Each region is colored in a shade of red, green, and blue, according to the proportion of votes received by each candidate. If the Republican, Independent, and Democratic candidates receive  $a_1$ ,  $a_2$ , and  $a_3$  votes, respectively. Formula to decide coloring of each region:

$$(R, G, B) = \left( \frac{a_1}{a_1 + a_2 + a_3}, \frac{a_2}{a_1 + a_2 + a_3}, \frac{a_3}{a_1 + a_2 + a_3} \right)$$



# Implementation

Reads latitude and longitude values from the.txt file of the region. Draws every sub region in the region with ACM GraphicsProgram according to their latitude and longitude values. For every sub region it reads the election results from the according txt file. Stores each sub region's information, election results and drawn shape in a Hash Map. This Hash Map's key is the shape (GPolygon object) and value is ElectionData object. This Hash Map is for GUI interactivity. Then the sub region is colored according to votes each party got.

Whenever user clicks on a sub region, the program shows information about the clicked sub region.

# Problems

- 1) Searching for every sub region in election result data List.
- 2) Showing a spesific sub region's information and election results whenever user clicks on a sub region.

# Problem 1 – First Solution: Linear Search

Time Complexity :  $O(n)$  for every sub region (theoritacally).

Because data mistakes it costs in average **4 times** time complexity.

Try to optimize it!

# Problem 1 – Second Solution: Sort Input

Merge Sort

Time Complexity: worst case  $O(n \log n)$  Space Complexity:  $O(1)$  for every region.

Then

Binary Search

Time Complexity: worst case  $O(\log n)$  Space Complexity:  $O(1)$  for every sub region.

But data mistakes didn't let it work as calculated 😞

Had to search several times if not found.

# Problem 1 – Third Solution: Linear Search

Data was already sorted. But not completely ☹️. Only ~90%.

90% is a high percentage for not sorting.

Do not sort the data. Only use binary search.

Time Complexity: Worst Case  $O(\log n)$  for no data mistakes.

Ignoring data mistakes.



# Problem 2 – Solution

Using hash map. Key as Gpolygon Value as sub region information.

Time Complexity:  $O(n)$  for every sub region clicked.

Space Complexity:  $O(n)$

# Total Complexity

$s$  = # of sub regions in a region

$n$  = # of regions

$v$  = # of coordinates

## Time Complexities

Drawing Map :  $O(v)$

Election Result Search:  $O(n \cdot \log s)$

User Click:  $O(s \cdot n)$

## Space Complexities

Drawing Map :  $O(s)$

Election Result Search:  $O(s)$

User Click:  $O(s \cdot n)$

Thank You For Listening