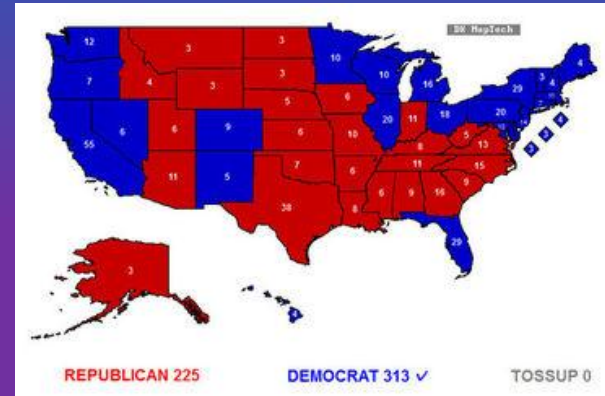


Purple America

Ogün ÖZ

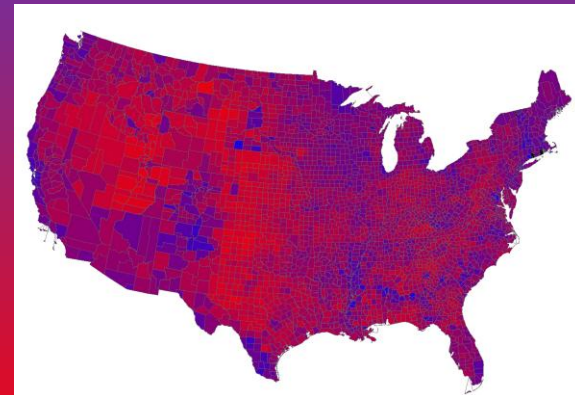
What is Purple America?

- An election map



What is different?

- Possibility to show approximately the voting percentages visually.



Possibility to apply Dynamic Programming!

- First draw and paint one district of a state.
- Solving sub-problems will lead to solve the global problems.
- «An optimal solution to a problem contains within it optimal solutions to sub-problems»

Useful for other data sets for America.

Ex: Education Level or Male-Female Gender Ratios in Districts or States.

Problem Definition

- **Input:** Coordinates and names of the districts (or states), list of election results per states and a set of desired region's coordinates.
- **Output:** Filled and painted polygons and a text showing election results for a region.
- Draw the polygons using the coordinates of the regions, and paint them according to the election result for that region using RGB colors. Show the election result for the polygon that is selected by the user.



Solution

- ElectionMap ADT: HashMap that can handle the string comparison problems.

s = # of string (the name of the region)

d = # of districts in a state

When inserting, it splits the key.

$O(s)$ expected, $O(d*s)$ worst case.

When searching, if it can't find, to lowercase, check word by word.

$O(1)$ expected, $O(d*s)$ worst case.

- Memorize just currently drawing state's election results in ElectionMap ADT.

$O(d)$ space complexity.

Step by Step

- Read the name of the state
- If it is a new state, insert election results for that state
- Draw the region(polygon)
- Search the results for that region in ElectionMap
- Paint the polygon

Showing the Election Results for the Desired Polygon

- Colors mean the voting percentages!
- However!

To know the NUMBER of democrat, republican and other votes, we need the total vote number for that region.

Unfortunately, for every polygon, the total vote number has to be memorized.

Analysis of Solution

s = # of string (the name of the region)

d = # of districts in a state

n = # of regions

m = # of states

v = # of coordinates

For every m state, insert and get election results temporarily.
 $O(m*s)$ expected $O(m*d*s)$ worst case insertion plus $O(m)$ expected
 $O(m*d*s)$ worst case searching. $O(d)$ space complexity!

For every polygon n , the total vote number should have to be memorized. $O(n*1)$ expected $O(n*n)$ worst case insertion. Searching number is up to user yet mostly less than insertion number. $O(n)$ space complexity!

- **Total time complexiy:** $O(v)$ for drawing polygons using v vertices + $O(n)$ expected + $O(n^2)$ worst case (insert total vote values for n polygon) + $O(n)$ for getting the desired polygon's election result + $O(m*d*s)$ expected + $O(m*d^2*s)$ worst case for inserting and searching $m*d$ election result.

$$v > n \geq m, d > s$$

$$\text{Approximately } m*d = n$$

Expected time complexity: $O(v)$

Worst case time complexity: $O(n^2)$

- **Total space complexity:** $O(d) + O(n)$, $n > d$
 $O(n)$

Possible Optimizations

- Change ADT to perform insert and search operations.
- Insert and search numbers are just slightly different.
- Find an ADT to perform both operations effectively.
- AVL Tree!
 - $O(\log(n))$ worst case, $O(\log(n))$ average time complexity.
 - It's better for asymptotic worst case complexity.
 - But it's a real world project, and I preferred to optimize expected running time.

Thanks for listening!