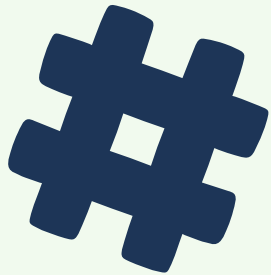


EC - 504 Project



Nearest State/Country Finder



Hitanshi Jain
Renyu Jiang
Shuyan Zhang
Yuxiang Wan

Project Mission



- Load the reference points in an efficient data structure.
- Allow users to input latitude and longitude into the data structure
- Return the K nearest reference points as output.
- Find the state and county of the input reference point by computing a majority voting among the 5 nearest points.

Github repo: https://github.com/hitanshijain/Nearest_State_Finder



Implementation



- For distance computation between two points, we can use the “equirectangular approximation”

$$x = (\lambda_2 - \lambda_1) * \cos((\phi_1 + \phi_2)/2);$$

$$y = (\phi_2 - \phi_1);$$

$$\text{Distance} = \text{Sqrt}(x*x + y*y) * R;$$

where ϕ is latitude, λ is longitude, R is earth's radius (mean radius = 6371 km);



A:

Construct an AVL

$O(N \log N)$

+

AVL find 10 smallest

worst case: $O(\log N)$

B:

Median Quicksort $O(N + N \log N)$ Even in the worst case

Doubted:

C: Best $O(2N)$ Worst $(N + N^2)$

1. Find the largest L and smallest S $O(N)$

2. Idea From Dictionary Search: Best $O(N)$ Worst $O(N^2)$

Array A in size of N

Smallest: $A[0] = S$

Biggest: $A[N-1] = L$

New element: $A[N * (NEW - S) / (L - S)]$

$Idx = N * (NEW - S) / (L - S)$

While ($A[Idx] \neq 0$ and $A[Idx] < New$) {

$Idx += 1$

}

While ($A[Idx] \neq 0$ and $A[Idx] \geq New$) {

$Idx -= 1$

}

(May have problem when the values are too closed)

$A[Idx] = New$

3. $A[0:10]$



Another Possible implementation:

C++ Map:

The key values are in increasing order by default

Searching time: $O(n)$

Insertion time: $O(\log(n) + \text{Rebalance})$

Deletion time: $O(\log(n) + \text{Rebalance})$

Key: Distance

Value: Index

```
idx = std::next(map, 10);
```



Thank you

