Interview Questions: Algorithms

Signal Data Science

- Write a function to calculate all possible assignment vectors of $2n$ users, where n users are assigned to group 0 (control) and n users are assigned to group 1 (treatment).
- Given a list of tweets, determine the top 10 most used hashtags.
  - For each tweet, split the string by spaces, extract substrings that begin with #, and convert them to lowercase.
- You have a stream of data coming in of size n, but you don't know what n is ahead of time. Write an algorithm that will take a random sample of k elements. Can you write one that takes $O(k)$ space?
  - This is reservoir sampling. Populate a list with the first k elements of the list. Then iterate through the rest; at the ith element, generate a random integer j between 1 and i inclusive; if $j \leq k$, then replace element j in the list of items with the ith item. Then the ith element of S is chosen to be included with probability k/i, and at each iteration the jth element is chosen to replaced with probability $(1/k) \cdot (k/i) = 1/i$.
- Write an algorithm to calculate the square root of a number.
  - Repeatedly average the estimate e with n/e.
- When can parallelism make your algorithms run faster? When could it make your algorithms run slower?
  - Answer on Quora
  - Parallelism works when you can subdivide your algorithm into a large number of independent parts. However, if each subdivision of the problem needs to communicate with the others, then parallelism will help much less. Combined with greater overhead (both computationally, from managing the different parallel threads, or on non-CPU hardware, if jumping from place to place in the hard drive slows the process down substantially), parallelism can make your algorithm run slower.
- Write functions to test if a string is a palindrome (the same both forwards

and backwards) and to test if a string contains a palindrome as a substring. Do the latter in $O(n)$ time.

  - Existence of a palindrome equals existence of a palindrome of length 2 or 3, so loop through once checking for two consecutive identical characters and then loop through another time with a "window" of 3 characters checking for palindromes.

- Suppose you have an even number of points in the 2D plane. Write a function to give the parameters of a line which divides these points into two equally sized groups.

  - I haven't done this myself, but maybe you can find the centroid of the points, pick an arbitrary direction radiating out from the centroid to be 0 degrees, and order the points according to the order in which an arc would "sweep out" the points; then look at the 1st and $(n/2+1)$th points and fiddle with the line a bit. This fails when you have collinearity in bad places relative to lines radiating outward from the centroid, so you'd have to pick a different direction for 0 degrees.

- Given a list of integers, find 3 different integers in the list which sum to 0.

  - This is the 3SUM problem. Iterate through the list, storing each number in a hash table. Iterate through every pair of integers and check if the negative of their sum is in the hash table. This solution is in $O(n)$.

- Given a list of integers, find the *continuous subsequence* with the largest sum.

  - This is the maximum subarray problem and can be solved in $O(n)$ time. Iterate through the list of integers, storing a partial sum as you go along; if the partial sum gets to 0 or below, start summing again with the next integer.