Interview Questions: Radix Sorts

Warning: The hard deadline has passed. You can attempt it, but **you will not get credit for it**. You are welcome to try it as a learning exercise.

☐ **In accordance with the Coursera Honor Code, I (Atul Gupta) certify that the answers here are my own work.**

Question 1

2-sum. Given an array a of N 64-bit integers and a target value T, determine whether there are two distinct integers i and j such that $a_i + a_j = T$. Your algorithm should run in linear time in the worst case.

Question 2

American flag sort. Given an array with N distinct values between 0 and $R-1$, rearrange them in ascending order in linear time and with extra space at most proportional to R.

Question 3

Cyclic rotations. Two strings s and t are *cyclic rotations* of one another if they have the same length and s consists of a suffix of t followed by a prefix of t. For example, "suffixsort" and "sortsuffix" are cyclic rotations.

Given N distinct strings, each of length L, design an algorithm to determine whether there exists a pair of distinct strings that are cyclic rotations of one another. For example, the following list of $N = 12$ strings of length $L = 10$ contains exactly one pair of strings ("suffixsort" and "sortsuffix") that are cyclic rotations of one another.

```
algorithms   polynomial   sortsuffix   boyermoore
structures   minimumcut   suffixsort   stackstack
binaryheap   digraphdfs   stringsort   digraphbfs
```

The order of growth of the running time should be NL^2 (or better) in the worst case. Assume that the alphabet size R is a small constant.

Signing bonus. Do it in NL time in the worst case.

☐ **In accordance with the Coursera Honor Code, I (Atul Gupta) certify that the answers here are my own work.**

Submit Answers Save Answers

You cannot submit your work until you agree to the Honor Code. Thanks!