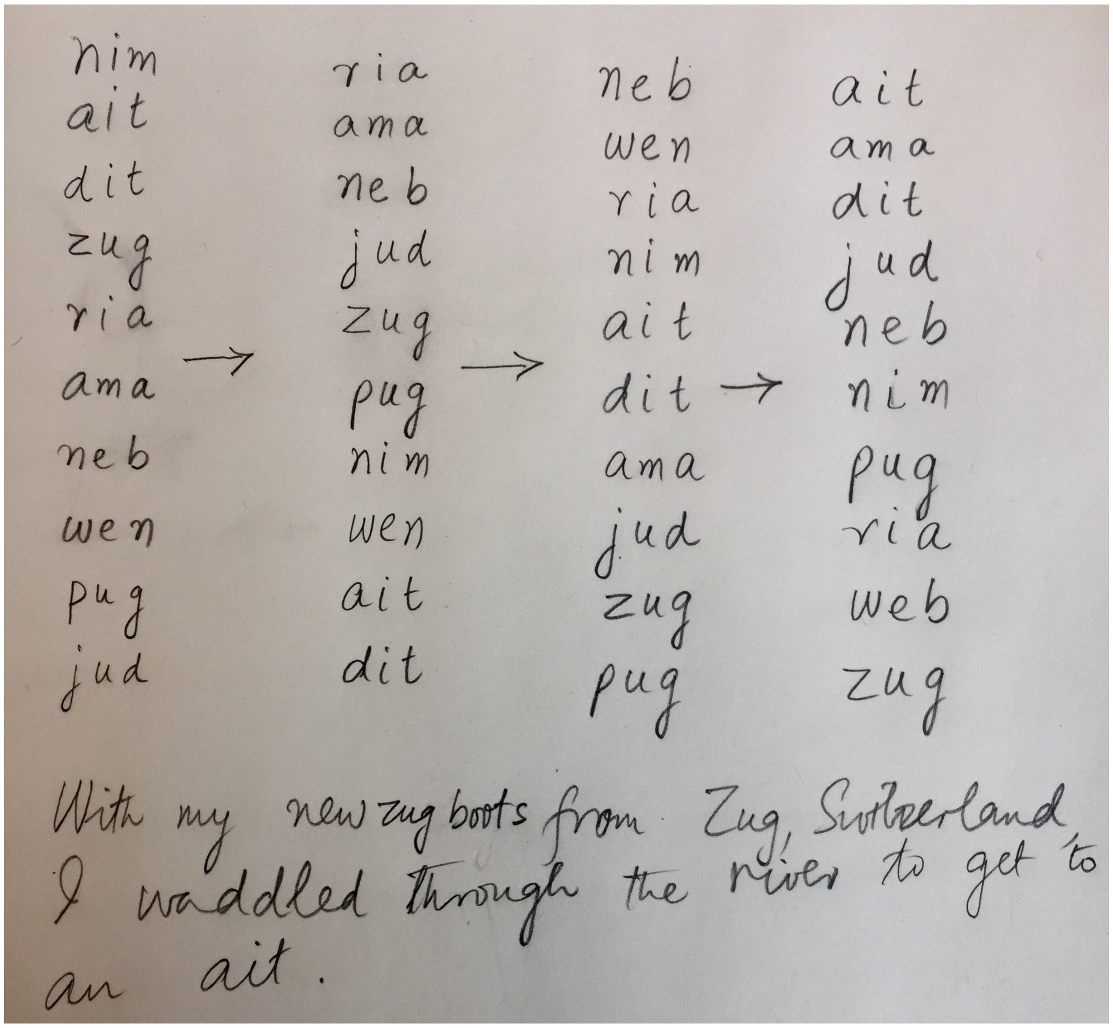
Problem 1. Illustrate the operation of Radix sort on the following list of English words: *nim*, *ait*, *dit*, *zug*, *ria*, *ama*, *neb*, *wen*, *pug*, *jud*. Form a meaningful sentence using the words *zug* and *ait*.

Solution:



nim		ria		neb		ait
ait		ama		wen		ama
dit		neb		ria		dit
zug		jud		nim		jud
ria		zug		ait		neb
ama	→	pug	→	dit	→	nim
neb		nim		ama		pug
wen		wen		jud		ria
pug		ait		zug		web
jud		dit		pug		zug

With my new zug boots from Zug, Switzerland I waddled through the river to get to an ait.

Problem 2. The runtime for radix sort is given by $\theta(d(n+r))$, where n is the length of the input array to be sorted, d, is the number of digits in each number and r is the Radix. Suppose, we want to sort an array of 4294967296, 128-bit numbers. Find the number of digits, d and compare radix sort with the runtime of Quicksort algorithm for the following two cases:

1. Radix of 4.

Solution:
$d = lg_r 2^b = lg_4 2^{128} = 64$

Radix sort runtime:
$\theta(d(n+R)) = \theta(64(4294967296+4)) = \theta((4294967296+4)\times 64)$

Quicksort runtime:
$\theta(nlgn) = \theta(4294967296 lg 4294967296) = \theta(4294967296 \times 32)$

Use quicksort. It has a better runtime than Radix sort and it is in-place.

2. Radix of 16.

Solution:
$d = lg_r 2^b = lg_{16} 2^{128} = 32$

Radix sort runtime:
$\theta(d(n+R)) = \theta(32(4294967296+16)) = \theta((4294967296+16)\times 32)$

Quicksort runtime:
$\theta(nlgn) = \theta(4294967296 lg 4294967296) = \theta(4294967296 \times 32)$

Radix sort and Quicksort are comparable in runtime. As the radix increases, radix sort performs better.

Which algorithm would you use for the two cases, respectively? Why?

Problem 3. We are given n points in the unit circle, $p_i = (x_i, y_i)$, such that $0 < x_i^2 + y_i^2 \leq 1$ for $i = 1, 2, \ldots, n$. Suppose that the points are uniformly distributed; that is, the probability of finding a point in any region of the circle is proportional to the area of that region. Design an algorithm with an average-case running time of $\theta(n)$ to sort the n points by their distances $d_i = \sqrt{x_i^2 + y_i^2}$ from the origin.

Solution: The total area we need to divide is π among n points
We will define the radius, r_i to be $\sqrt{\frac{i}{n}}$ and the regions for each coordinate (x, y) lie within the radii, $r_{i-1} \leq x^2+y^2 \leq r_i$. These regions represent the buckets. The points are uniformly distributed and each of these regions has equal area. Now we can apply bucket sort which has an average runtime of $\theta(n)$.