\text{BucketSort} \ (A, m) \ // \text{Sorts list } A \text{ where each value in } A
\text{count}[0..m-1] = 0 \ // \text{is in range } [0,m-1].
\text{for } i = 0 \text{ to } n-1
\qquad \text{count} [A[i]]++;
\text{for } i = 0 \text{ to } m-1
\qquad \text{print } i \text{ count } [i] \text{ times}.

\text{Time: } O(mn)

\text{Bucket sort seems very restrictive (only integers)}.

\text{How to sort:}

\begin{itemize}
  \item 0.78, 0.17, 0.39, 0.26, 0.72, 0.94, 0.21, 0.12, 0.23, 0.65
\end{itemize}

\begin{tabular}{|c|c|}
  \hline
  0 & \text{\texttt{[}0.17, 0.12\texttt{]}} \\hline
  1 & \text{\texttt{[}0.26, 0.21, 0.23\texttt{]}} \quad \text{Now sort each bucket} \\hline
  2 & \text{\texttt{[}0.39\texttt{]}} \quad \text{with insertion/quick sort} \\hline
  3 & \text{\texttt{[}0.68\texttt{]}} \quad \text{\# items in bucket will be } O(1) \\hline
  4 & \text{\texttt{[}0.78, 0.72\texttt{]}} \quad \text{(since elements will be evenly distributed)} \\hline
  5 & \text{\texttt{[}0.94\texttt{]}} \\hline
\end{tabular}

\begin{equation}
\frac{\text{total \# elements}}{\text{\# buckets}} \sim \frac{n}{c}
\end{equation}
Generalized Bucket Sort - distribute elements into an array of "buckets". Then, sort each bucket.

Can skip this example.

Radix Sort: A list of fixed-length integers.

Start with least significant digit.
Continuously sort by next least significant digit.

Ex: 170 45 75 90 2 24 802 66

170 045 075 090 002 024 802 066

170 090 002 802 024 045 675 066

002 802 024 045 066 170 075 090
Radix Sort

Suppose sorting fixed-length strings alphabetically

cats
cost
cast
cuts
coot

How?
Sort letter-by-letter starting from first letter

cats  cats  coat  cats
cost  cost  cost  cuts  <not working...>
cast  cost  cost  coat
cuts  coat  cats  cast
coot  cuts  cuts  cost

Problem: when we sort by letter to the right, we "unsort" previous letters.
Instead, start with last letter

cats  cats  coat  cost
cost  cuts  cost  cats  
cast  cost  cast  coat
cuts  cast  cats  cost
coot  coat  cuts  cuts
Why does this work?

Now, if we "unsort" letters to the right, it's ok since letter to the left is more important in the sort.

Radix Sort: A is list of fixed length Strings, integers

Start with least significant character, digit

Continuously sort by next least sig. character, digit.

What if not fixed length?

caster
cats--
coats--
costly

cuts--

Add a character that is alphabetically => coats-

before 'a' to the end: costly

cuts--
caster cats--
coats--
costly caster
cuts--
caster cats--
coats--
caster costly coats--
cuts--
caster costly cats--
cuts--
cuts--

sorted!
Run Time for Radix Sort?

Notice: for each character, we had to sort the elements by that character. Can use Bucket Sort!

Since we know range of values.

Therefore, runtime depends on:

\[ n: \text{# elements (Strings, integers)} \]
\[ d: \text{# characters/digits in each element} \]
\[ m: \text{# possible values for each character/digit} \]

Strings:
\[ m = 26 \text{ (or 27 if we must append "}")} \]

Integers:
\[ m = 10 \]

Total Time? For each character, perform bucket sort.
\[ O(d(m+n)) \]