Dictionary ordering on strings (lexicographic ordering)

Eg. cave, bat, at, a, ba, bat cave, ba,

Prefix of word appears before

Can we use radix sort for lexicographic ordering?

What difference does it make if we sort from LSB → MSB vs. MSB → LSB

Try to come up with "inefficient" examples.

Obs. By appending blanks at the end of the word, we can apply normal radix sort to strings (right to left)
First round

bat at at
bat at at
cave cave cave

Idea: Involve only those words in round i where the words have non-blank characters in i're position for right

Obs: If a string has length $l_i$ then it will be involved in $l_i$ rounds of bucket sort

$\Rightarrow$ Cost of sorting $= \sum l_i$ $\leq \sum m_j$ where $m_j$ is the number of strings having non-blank characters in j'th round

$\sum_{j}(m_j + 2 \cdot b)$

We want to implement bucket sort where $#$ buckets can be ignored and we know in advance the non-empty buckets
If the non-empty buckets sequence is known then cost of bucket sort (count sort) is $O(n)$ where $n$ is the # of elements.

$cavc: (1,c^1) (2,a^1) (3,v^1) (4,c^1)$

$bat: (1,b^2) (2,a^2) (3,t^2)$

$at: (1,a^3) (2,t^3)$

Let us apply sorting in the triples

$(1,a^3) (1,b^2) (1,c^1) (2,a^1) (2,a^2) (2,t^2) (3,t^2) (3,v^1) (4,c^1)$

Sorting can be done in time proportional to $t \leq L_i = O(N) \left[ O(l+N) + O(\Xi_1+N) \right]$.

(we are ignoring the cost of buckets or we are assuming that #buckets $\leq$ #elements)

We are not paying the cost for blank char

$\leq: \text{ alphabet}$