

Reverse a Sentence

Given a pattern of words separated by spaces (e.g. "bob likes bananas"), reverse the order of the words (e.g. "bananas likes bob").

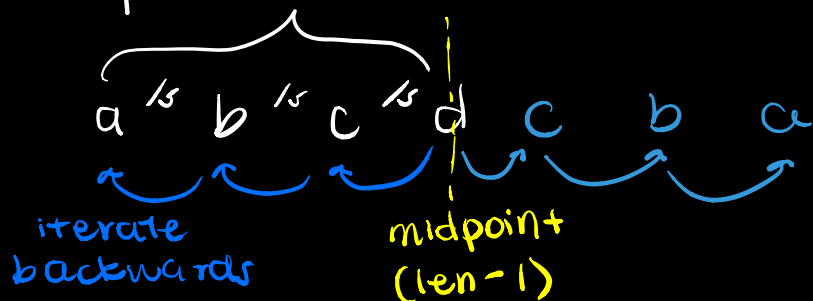
Thoughts:

Reversing strings reminds me of palindromes, because palindromes contain reversible substrings.

r a c e c a r \rightarrow race & ecar

If we tokenize the sentence, splitting at whitespaces & making the words our tokens, then we could iterate the sentence backwards & append the words in reverse order.

each letter here
represents a word



as we iterate backwards,
append each word to
the end of the original
sentence.

Then, we can chop off the original sentence, minus the last word.

Alternatively, instead of appending to the original sentence, we can just create a new variable to store the reversed sentence.

However, this solution requires a space complexity of $O(n)$, where n is the total number of characters of the input.

A solution that has constant space:

Strings are arrays, with each character being accessible as an element of the array.

Consider swapping in place—we would have to do it by letter, not entire words.

b	o	b	/s	l	i	k	e	s	/s	b	a	n	a	n	a	s
---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---

1 swap

s	o	b	/s	l	i	k	e	s	/s	b	a	n	a	n	a	b
---	---	---	----	---	---	---	---	---	----	---	---	---	---	---	---	---

finish swapping at $\frac{1}{2}$ (length) of the array

s	a	n	a	n	a	b	/s	s	e	k	i	l	/s	b	o	b
---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---

The entire array is now reversed. However, each word is also backwards.

To solve this, we can just continue to reverse letters, with the white space acting as our partitions.

b	a	n	a	n	a	s	/s	s	e	k	i	l	/s	b	o	b
---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---

:

b	a	n	a	n	a	s	/s	l	i	k	e	s	/s	b	o	b
---	---	---	---	---	---	---	----	---	---	---	---	---	----	---	---	---

And our result is "bananas likes bob"

Complexity

The optimal solution has $O(1)$ space complexity & uses temporary variables to perform swapping.

Time complexity is $O(\frac{1}{2}n + \frac{1}{2}n) = O(n)$

The first time we reverse the entire array we only iterate to the halfway point ($\frac{1}{2}n$). We also have to reverse each word using the same method, the iteration also amounts to $\frac{1}{2}n$. Therefore, the time complexity is linear.