

## 1. Gap Buffers

### Data Structure Description

The gap buffer data structure is commonly used by text editors for storing the files that are currently being edited. The data structure is simply an array that stores the contents of a file plus some extra empty space. This empty space is called the gap. The gap is moved to the position where the user is currently editing text (the location of the cursor). In addition to the array, the gap buffer data structure stores the location and length of the gap. For example:

The [            ]fox jumps over the dog.

The square brackets denote the beginning and end of the gap. In this example the gap has a length of 10. Let's type quick :

The quick [     ]fox jumps over the dog.

The length of the gap is now 4. Let's type brown :

The quick brow[ ]fox jumps over the dog.

The after entering the first 4 characters the length of the gap is 0, so a larger array needs to be allocated for the gap buffer:

The quick brow[            ]fox jumps over the dog.

The rest of brown can now be entered:

The quick brown [            ]fox jumps over the dog.

The backspace operation is done by moving the start of the gap backwards. Let's delete brown:

The quick [brown            ]fox jumps over the dog.

Let's now type red :

The quick red [n            ]fox jumps over the dog.

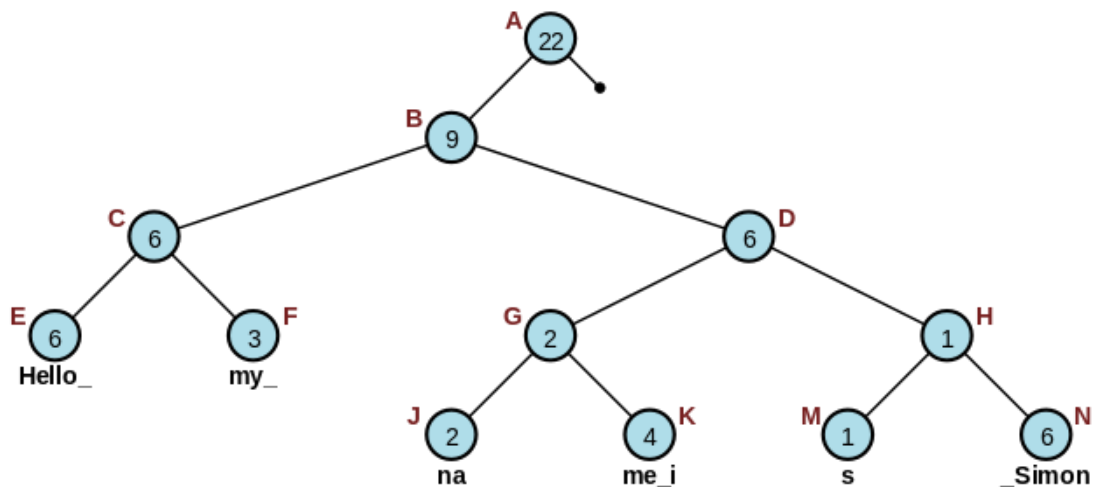
The cursor is moved by relocating the gap, this is done by copying characters from one side of the gap to the other. Let's move the cursor before the word "dog":

The quick brown fox jumps over the [n            ]dog.

### Problems

1. What is the run time complexity of inserting a character, when the length of the gap is at least 2?
2. What is the run time complexity of doing a backspace?

3. What is the run time complexity of moving the cursor forward  $n$  characters? Backward  $n$  characters?
4. What is the run time complexity of entering 8 characters into a gap that is only 4 characters long? Assume the array storing text is length  $n$  and the size of the newly allocated array will be length  $n + 10$ . Neglect the time it takes to allocate new memory.
5. Some text editors use the rope data structure for storing text buffers. This data structure was briefly discussed during the lectures on binary trees. The rope data structure stores the text in a binary search tree. The search tree can be used to find the character at an index in the buffer:



Insert operations involve splitting the node where the cursor is and inserting a new node for the new text. Delete operations involve splitting the current node and removing the node that contains the text to be deleted. Compare and contrast the insert and delete operations on the rope data structure and the gap buffer data structure.

## 2. $k^{\text{th}}$ element

Develop an algorithm that computes the  $k$ th smallest element of a set of  $n$  distinct integers in  $O(n+k \log n)$  time.