Solutions

Problem 1

1. θ(1)
2. θ(1)
3. Forward: θ(n) Backward: θ(n)
4. θ(n) (need to copy the n characters from the old buffer into the new buffer)
5. Gap Buffer is more efficient for insertions and deletions at the current location of the cursor. Rope is more efficient for jumping to an arbitrary location in the file and doing an insert/delete. A jump and insert/delete takes θ(n) time for a gap buffer and θ(ln(n)) for a rope.

Additional Notes:

- In depth explanation of rope data structure
- Gap buffer is easier to implement than ropes
- Searching for text is easier more efficient in a gap buffer

Problem 2:

Firstly, build a min heap – that takes O(n). After that, pop from the top k times – popping and maintaining the heap takes O(logn) for every time we pop, therefore overall complexity becomes O(n + k logn).