In lab exercises
Try to design the following basic I/O efficient data structures. There are two primary ways of viewing blocks. First, imagine a set of blocks as a contiguous array on disk, like a single file. Alternatively, Each block can be like a node in a graph with multiple links to other blocks.

1. Design an I/O efficient stack, where $N$ push/pop operations take at most $N/B$ I/Os.

2. Design an I/O efficient queue with the same bounds as the stack. Is your implementation array based or node based? Can you design an solution both ways?

3. Design a structure to efficiently search a sorted collection of data. What do you think an efficient bound would be?

4. Can you make your structure in the previous question dynamic, so that it can support insertions and deletions? What are the run-time bounds of these updates?