1. Prove that in Van Emde boas layout of n numbers, searching a number needs at most $4\log_B(N/B)$ I/Os. If we search all numbers existing in the tree in an arbitrary order, prove a search needs $2\log_B(N/B)$ I/Os in average.

2. Analyze a B-tree over N numbers when it is constructed using the Logarithmic Method. Assume there is only insertion.

3. Consider the algorithm that computes the average value in an $m \times m$ array A column by column, for the case where A is stored in row-major order. The algorithm performs n I/Os, where $n := m^2$ is the total size of the array, because whenever we need a new entry from the array we have already evicted the block containing that entry. This is true when the Least-Recently-Used replacement policy is used and when $m > M/B$, but it is not immediately clear what happens when some other replacement policy is used.

   (i) Suppose that $m = M/B + 1$, where you may assume that M/B is integral. Show that in this case there is a replacement policy that would perform only $O(n/B+\sqrt{n})$ I/Os.

   (ii) Prove that when $m > 2M/B$, then any replacement policy will perform $\Omega(n)$ I/Os.

4. A *coloring* of an undirected graph $G = (V, E)$ is an assignment of colors to the nodes of G such that if $(v_i, v_j) \in E$ then v_i and v_j have different colors. Suppose that G is stored in the form of an adjacency list in external memory. Assume the maximum degree of any node in G is $d_{\max}$. Give an algorithm that computes a valid coloring for G that uses at most $d_{\max} + 1$ colors. Your algorithm should perform $O(sort(|V| + |E|))$ I/Os.

5. A shortest path data structure for a graph G is a data structure which, given two query vertices x and y reports the distance (and the corresponding shortest path) from x to y. In general, such data structures are difcult to construct, at least if we want the data structure to be small and answer queries quickly. Now assume the given graph is an unweighted tree. Develop a data structure that uses linear space and can report the distance between any two vertices using $O(\log_B N)$ I/Os and report the path in $O(1 + K/B)$ I/Os where K is the length of the shortest path.