Implementation of DBMS
Exercise Sheet 6
Klingemann, WS 2025 / 2026

1) Use the same scenario as in task 4 of Sheet 5. We assume that nothing is in memory initially, and that the search key is the primary key for the records. What is the average number of disk I/O's needed to retrieve a particular record that is stored in our data file given its search key using the following approaches?
a) We do not use any index but sequentially inspect the data file from the beginning until we find the record.
b) We use the dense index described in task 4b to retrieve the record. We sequentially scan the index from the beginning until we find the search key value we are looking for.
c) We use the multi-level index described in task 4d to retrieve the record.
d) We use the multi-level index described in task 4e to retrieve the record.

2) Suppose that blocks can hold either ten records or 100 key-pointer pairs. We have a data file that is a sequential file and a sparse index on this file. The index has multiple levels up to a level with just one block. Each primary block of the data file has one overflow block. The primary blocks are full, and the overflow blocks are half full. However, records are in no particular order within primary block and its overflow block. All index blocks are 60% full.
a) Calculate the total number of blocks needed for a 3,240,000-record file and the index.
b) Calculate the average number of disk I/O's needed to retrieve a record given its search key by using the index. You may assume that nothing is in memory initially, and that the search key is the primary key for the records.

3) Suppose that blocks can hold either three records, ten key-pointer pairs, or fifty pointers.
a) We use the indirect bucket scheme. If each search-key value appears in 10 records, how many blocks do we need to hold 3000 records and its secondary index structure? How many blocks would we need if we did not use buckets but a key-pointer pair for each record?
b) We want to retrieve all records for a particular search key. In doing so, we sequentially scan the key-pointer pairs in the index from the beginning until we find the search key value we are looking for. We assume that records with the same search key never share a block. We also assume that buckets never extend across different blocks and that key-pointer pairs with the same key are all in the same block. How many blocks do you have to inspect on average, for the two scenarios in a)?

4) What is the minimum number of keys in B+-tree (i) interior nodes and (ii) leaves, when
a) $n = 10$
b) $n = 11$