1 General External Merge Sort

Pass 0 – Use B buffer pages. Produce N/B sorted runs of B pages each.
Further passes – Merge B-1 runs.

(a) You have 4 buffer pages and your file has a total of 108 pages of records to sort. How many passes would it take to sort the file?

(b) How many runs would each pass produce?

(c) What is the total cost for this sort process in terms of I/O?

(d) If the pages were already sorted individually, how many passes would it take to sort the file and how many IOs would it be instead?

(e) Given an arbitrary N number of buffer pages, if we could sort the file in 3 passes, what order of magnitude is the size of the file in terms of number of pages?

2 Hashing

(a) If we had records that we wanted to hash where the key being used were random integer values distributed uniformly, would the length of the integer be a good hash key? Why?

(b) What are some of the use-cases of hashing is preferred over sorting?

(c) We can process B × (B-1) pages of data with external hashing in two passes. For this case, fill in the blanks with the appropriate number of pages, where we have B pages of available RAM (buffer pages).
______ input buffer(s)
______ partitions after pass 0
______ pages per partition

(d) If you are processing exactly B × (B-1) pages of data with external hashing, is it likely that you'll have to perform recursive external hashing? Why or why not?

(e) Can we use the same hash function for partitioning (divide) and rehashing (conquer)? How about for the recursive partitioning?