Data structures and algorithms

Consider a hash table with S entries that is used to store N records, where S>N.

a) Under the assumption of perfect hashing, what is the algorithmic complexity (using O() notation) of inserting a new element to the table? What is the complexity of hash table lookups? Explain your answer.

b) Under the assumption of imperfect hashing function, what is the worst-case complexity of hash table lookups? Explain your answer.

c) Describe two techniques that can be used to handle collisions during hash table insertion, and discuss their relative advantages and disadvantages. Illustrate with an example.

Operating Systems

Modern microprocessors and multi-task operating systems support the abstraction of virtual memory with the aid of page table data structures. Consider a processor with physical memory addressable using 32 bits and a 40-bit virtual memory address space. The processor supports paging with a fixed page size of 64Kbytes, and protection mechanisms allowing for each page to have read, write and execute permissions set.

a) What is the minimum size (in bits) of a page table entry (PTE) for this system? Justify your answer, describing what information needs to be stored in a PTE.

b) Let the size of a PTE be a constant "S". What is the minimum size of the page table for an O/S process if a "flat" page table is used? What is the minimum size if a two-level page table is used? (Assume 14 bits index the first level and 12 bits index the second level of the table).

c) Assume the time is takes for the processor to read a PTE from memory is 100ns, and the time to look up a translation look-aside buffer (TLB) entry is 1ns. What TLB hit rate is necessary to obtain an average virtual-to-physical address translation time of 2ns?

d) Explain three advantages and one disadvantage of using virtual addresses compared to physical addresses.