CS EE Reflection

Topic - Comparison of search algorithms in the context of organised file systems

RQ - To what extent is Quadratic Search more efficient than Jump Search with optimal step in the context of sequentially organised files which are sorted by a single index in file management with Binary Search as a benchmark for efficiency.

File management systems are an important part of all types of computer systems worldwide. They are not only used in Operating Systems like Windows, Mac, Linux, etc., but also in server-based web applications like Netflix or Amazon. As such, searching for files in such a system is an important component and optimal efficiency must be achieved. In unordered file systems, a linear search may be preferred. However, in sequentially organised file systems, depending on context, various types of searches may be optimal.

My inspiration for the broader topic “*Comparison of search algorithms in the context of organised file systems*” was MacOS’s search tool Spotlight. Spotlight is a system-wide desktop tool that searches through all the files in the system in a matter of seconds by file type, name, and even location! Upon further research, I discovered that Spotlight is a selection-based search system, which creates an index of all items and files on the system, which means it likely uses some of ordered search. I wondered what exact search algorithm is used by Spotlight but could not find any answers as it is likely proprietary Apple software.

While I was researching Spotlight, I stumbled across some unusual ordered-search algorithms: quadratic search and jump search. These are both useful in their own contexts, however, inspired by Spotlight, I wanted to discover how they would hold up against not only each other, but also against the tried-and-tested binary search algorithm in the context of organised file systems. This is how I arrived at my final research question. Jump search is an algorithm that, while usually less efficient than binary search, might work better than binary search in some contexts. For example, in a context where jumping backwards takes significantly more time than jumping forward, jump search will be more efficient than binary search because jump search only needs to jump backwards once, while a binary can jump backwards up to log *n* time. Quadratic search is a generalisation of the binary search algorithm which worst case complexity O(log n). This New algorithm Quadratic Search has worst case complexity O(log n/2). However, again, this is less efficient in some cases compared to Jump Search. The objective of this paper will be to compare these two algorithms in the context of an organised file system with binary search as a benchmark for efficiency.