Queries with poor effectiveness or efficiency

1. of 1 and the to in
2. the and to in of
3. 1 2 3 4 5 6 7 8 9 10 20 25 50 100
4. of the a an in
5. if the course I want to take is not on the list, what should I do
6. course details for a software development program
7. how can i find the information that is on the page if it is not at the top of the search results
8. what can i do if the computer i use does not load the course page anymore
9. what is the page that has the program detail I need for my class schedule
10. the of and to in a by for at is on from

These queries in our initial implementation of the search engine all had major issues, including runtimes ranging from 0.5 - 3.0 seconds. The major change that improved the performance of the search engine was changing the format of the inverted index from a JSON-based implementation to a txt one. This gave us the ability to use a mmap to keep track of where in the index each term was, allowing us to use seek and prevent the hassle of having the entire inverted index loading into memory. The other major optimization that allowed us to get query times under the 300 milliseconds benchmark was sorting queries in order of their document frequency, filtering out more documents per binary AND operation.

Queries with fast runtime performance and consistent tf-idf ranking

11. cs 121 course information
12. python search engine
13. thornton class information
14. video game development
15. java courses spring
16. machine learning
17. ics career fair
18. recursion lecture
19. algorithms and design
20. principles of system design