

EECS 498 System Design of a Search Engine
Winter 2026 Midterm

This exam is closed book, closed notes, no cellphones, computers or other electronic devices. The standard amount of time for this exam is 80 minutes.

You are to abide by the University of Michigan College of Engineering honor code. To receive a grade, you must sign below, agreeing to abide by that code.

I have neither given nor received aid on this exam, nor have I concealed any violations of the Honor Code.

Signature _____

Printed name _____

Uniqname _____

This exam has 9 pages. Please be sure you have them all and that you have put your uniqname on every page.

1. In July 1974, in "The UNIX Time-sharing System", Dennis Ritchie and Ken Thompson wrote, "Perhaps the most important achievement of UNIX is to demonstrate that a powerful operating system for interactive use need not be expensive." UNIX could run on hardware costing as little as how much? (Adjusting for inflation, \$100 then had the same buying power as \$660 today.)

2. What are overlong sequences and why are they prohibited? Can you give an example and why it might be a problem?
3. Word locations can be numbered from (a) the start of the document or (b) the start of the index. Which is better and why?

4. You read an HTTP page with `LinuxGetUrl` and it returns a redirect to an HTTPS page but it's the wrong one. But if you type the same HTTP URL into your browser, it works. What's going on and why? How does this relate to your project?
5. You observe that `"LinuxTinyServer 5000 ."` works. But `"LinuxTinyServer 80 ."` fails. Why is that?
6. An HTML form might use GET or POST to return what's been typed by the user. What's the difference, is one better than the other for a search engine webpage, and why?

7. Suppose you have developed a formula for calculating static rank as a linear combination, $S = \sum a_i x_i$, where the x_i represent various heuristics, e.g., short URL, and the a_i represent the relative weights. Is it worth writing the S values into the index to save time recalculating them when you need them later?
8. Once you have your engine working on a single machine, what changes will you need to make to distribute it across multiple machines?
9. When a Linux process forks, the parent and child initially share their memory pages. But if one of them makes a change, what happens? What's that called?

10. When a process forks, how does it know if it's the parent or the child?
11. You have two threads sharing two resources, A and B. To use a resource, you must first lock it, then release the lock when you're done. Thread 1 needs A and sometimes B as well, so it locks A and then, when it needs it, it locks B also. Thread 2 needs B and sometimes A, so it locks B and then, when it needs it, it locks A. Will this work? If there's a problem, what must be done to fix it?
12. List the steps to reading a webpage using HTTP?

13. To serve up pages over HTTP, what does a simple web server do?

14. How do you set up a memory-mapped file?

15. What is the advantage of a memory-mapped file?

16. If you would like to be able to search for "amazon" only in the title in an inverted word index, how would you do that? If you do it that way and you now want to find pages that have "amazon" anywhere, what would you have to do?

17. Why would you need a seek index or synchronization table for a posting list?

18. What happens if you try to seek way past the end of a file and begin writing?

19. What is the difference between precision and recall? Which is more important for a search engine and why?

20. What is the difference between static rank and dynamic rank?

21. You can create a thread using `pthread_create()`. But what does `pthread_join()` do?

22. What is a sparse file?

23. What types of search engine queries are likely to be slowest? Would lots of words in the query make the search faster or slower? Why?

24. Would it be okay if errors in the HTML cause your search engine to misparse a page?

25. If you look up the IP address of a host, then connect to it, why do you still need to specify the host again in a GET request? You already know it's the right machine based on the IP address.