

EECS 498 System Design of a Search Engine
Winter 2026 Midterm **Solution**

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 unickname on every page.

Twenty-five questions, each worth 4 points for a *complete* answer.

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.)

\$40,000 in 1974. (Equivalent to \$264,000 today.)

2. What are overlong sequences and why are they prohibited? Can you give an example and why it might be a problem?

An overlong sequence is a UTF-8 character encoding that uses more bytes with extra high order zeros than is needed. They're prohibited as security risks. For example, encoding a / as a two-byte overlong character has been used to bypass security filters that only check for the standard one-byte encoding.

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?

It's better to go with start of index because the deltas when crossing document boundaries will be smaller and will need shorter UTF-8 encodings. (If you go with start of document, the high order document bits changing causes a large delta.) This is important because search time is linear with index size.

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?

Your browser isn't bothering to try the HTTP version. It's automatically trying the HTTPS version first and only falling back to HTTP if HTTPS doesn't work. They probably assume most sites now support HTTPS but that there are a lot old links out there that still say HTTP. It might not be a bad idea for your crawler to do this as well.

5. You observe that `"LinuxTinyServer 5000 ."` works. But `"LinuxTinyServer 80 ."` fails. Why is that?

Under Linux, ports 0 to 1023 are privileged. Only processes running with root (superuser) privileges can bind to these ports. This is to prevent unprivileged users from creating a malicious HTTP port 80, HTTPS port 443, or other server masquerading as a legitimate system application.

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?

GET returns whatever's been typed as a query string in the URL. Everything after the `"?"` is the query string, giving name=value pairs separated by ampersands. POST returns the same information but as content in the HTTP message. GET is better for a search engine because you can give the URL to someone else and they can get the same result without having to retype the query into the form.

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?

No, this is not helpful. You should store only raw data, e.g., URL length, not a heuristic test like short URL, and definitely not the S values. You'll want to experiment and revise your weightings and your heuristics to tune your ranker once everything is running, instantly making any S values stored in your index out of date. And anyway, calculating a simple sum of products is fast.

8. Once you have your engine working on a single machine, what changes will you need to make to distribute it across multiple machines?

When the front end gets a query, it should broadcast it to the sibling machines and then merge what they send back with the local results.

When the crawler finds a URL, it should decide which machine gets it, e.g., by hashing the URL modulo number of machines. It should crawl only its own slice of the web and send URLs in the other slices to the appropriate 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?

The shared pages are initially marked read-only. When a process tries to write to a RO page, that causes a hardware interrupt into the operating system, which makes a copy of the page, marks them both RW and gives one to the parent, the other to the child. This strategy is called copy-on-write.

10. When a process forks, how does it know if it's the parent or the child?

The return value is zero to the child and the child's PID (process ID) to the parent. The parent can then wait on the child PID for the child to exit and retrieve its exit status.

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?

No, this will not work. It will deadlock. To fix the problem, if multiple resources are needed, they must always be requested in the same order. Here, it must be either A, then B, or B then A. If the order is A then B, then for thread 2, already holding B, to acquire A, it must first release B, then lock A, then B again. Not technically required but considered good practice is releasing them in reverse order to that in which they were taken.

12. List the steps to reading a webpage using HTTP?

1. Determine the IP address and port number for the server.
2. Create a socket.
3. Connect it to the server.
4. Create an appropriate GET message for the page you want.
5. Write the GET message to the host over the socket.
6. Read the response and keep reading until there is no more.
7. Close the socket.

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

The server creates a socket, binds it to its IP address and port number, and begins listening. It then loops, accepting all connect requests, passing each request off to a separate thread to handle the expected GET request.

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

Open the file, then pass the file handle or descriptor to the operating system function `mmap( )` asking that it be mapped into your address space. It returns a pointer to the beginning of where it's been mapped.

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

There's no need to do individual file reads or writes to/from buffers. Instead, the whole file is mapped into the process virtual memory address space. To read/write any location in the file, all you have to do is dereference a pointer into that address range where the file has been mapped. When you read or write addresses in that range, the paging system is used to fetch pages from disk as needed. The file system caching mechanism will try to keep the most used blocks in memory without us having to do anything to make that happen.

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?

You could decorate the term with characters you expect to strip out from anything a user would type during query parsing to distinguish the posting list for amazon in each of these four cases. You get to pick any characters you like as long as they're characters you expect to strip. For example, you might index "@amazon" for occurrences in anchor text, "=amazon" for URL, "+amazon" for title and unadorned "amazon" for body. If you do this, then to search for "amazon" anywhere, you would have to OR (merge) the individual posting lists.

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

The posting list is typically compressed both by storing deltas from the previous post rather than absolute locations and by a variable-length encoding scheme, e.g., UTF-8. To know the absolute location of post, you have to add up all the deltas starting from some known location, either the beginning of the index or an entry in a seek index. The seek index could be an array, perhaps indexed by high bits of a seek address, or a list of known locations. Each entry is a pointer to a post, if there is one, and the location where it occurs.

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

The operating system will let you do it, possibly creating a "hole" in file where there's no data.

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

Precision is the fraction of the results that are relevant. Recall is the fraction of all relevant results that are returned. Precision is much more important for a search engine because no one ever wants to see all the matching pages, only the best.

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

Static rank is a measure of how good the page is without regard to the query.
Dynamic rank is a measure of how good a match the page is for the specific query.

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

`pthread_join()` waits until the specified thread ends and retrieves the exit status.

22. What is a sparse file?

A file with a "hole" in it where there's no data. Reading from the hole returns zeros.

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?

The slowest will be the one-word queries for common words, e.g., a query for "the". They'll match an enormous number of documents, forcing each to be ranked. Adding more words to the query, especially if some are rare, will make the search faster because it won't match as many documents, meaning there won't be many that need to be ranked.

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

Yes, if the HTML was broken, the page was probably lower quality anyway. If you miss an occasional word or link, especially because of broken HTML, it's unlikely that will affect search results in a noticeable way.

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.

Any given machine might be hosting several websites at the same IP address and port number.