Practice Final Exam

Comp 582 MCS@Rice Fall 2022
Time Limit: 5 Hours

Instructions

For the exam, you may use or refer to videos, video slide sets, problem sets, live session slides, and any of your personal notes or materials. You may use any result or theorem that was stated in a course problem set. You may also assume that any problem solved with an algorithm from the problem sets is correct. (You do not need to have gotten the problem right). You may *NOT*, however, collaborate with any other people.

You may also use the Internet in a limited way. You may search for something on the internet, but you may *NOT* ask an exam question on any question answering site (like Stack Overflow). Also, if you use something from the internet, you may **not copy it verbatim.** You must paraphrase into your own words. For code, please change a variable name and/or rearrange some of the serial lines**.** Furthermore, if you use something you found on the internet, please cite whatever you found.

Please show *all* relevant work, and *only* relevant work.

If you turn in spurious work, or random discarded approaches, I will assume that you think whatever you wrote should contribute to the solution. Anything you turn in will be assumed to be part of your answer.

As with your problem sets, you may turn in PDF or plain text. You must, however, turn in **only 1 file**

Also, do not forget to put the Honor Code Pledge on your last age of your exam (you may use a separate page if necessary). Please be sure to include your start time and end time as part of the Pledge.

When you have finished your exam, please upload to Canvas (just like the problem sets).

General Guidelines

- Asking for the "complexity" of an algorithm means the worst-case *time* complexity of the algorithm.
  - ✦You might be asked specifically for the "space complexity", or "time complexity", but absent the qualifier, "time complexity is the default
- If you are asked to give an "efficient" algorithm, then you should produce an algorithm with the best worst-case time complexity.
- Moreover, "worst-case" means the *best* worst-case behavior.
- By default, complexity and solutions to recurrences may be given in Big Oh notation.
  - ✦You may be asked for *exact* solutions in some cases. For those cases, Big Oh notation is ***not*** acceptable.

Short Answer Questions:

1. What is the worst-case complexity for each of these algorithms? (Use Big Oh notation):

    1.1. Selection Sort of N items

    1.2. Insertion Sort of N items

    1.3. Shell sort of N items

    1.4. A sequence of M union-find operations on N disjoint sets

    1.5. The copying cost when using the doubling algorithm for a sequence of N push operations.

    1.6. Quicksort of N items

    1.7. Merge sort of N items

    1.8. A LSB radix sort over an alphabet of size A, for fixed-length strings of length L. there are N such strings.

    1.9. Given a sample string of length N, and a fixed pattern (NOT a regular expression) of length M, what is the complexity of searching the sample string for the pattern?

    1.10. Given a sample string of length N, and a (fully parenthesized) regular expression of length M, what is the complexity of searching the sample string for the pattern?

    1.11. Find the maximum item in a max heap with N items

    1.12. Finding an item in a random binary search tree of size N.

    1.13. Finding an item in a left-leaning red-black tree of size N.

    1.14. Inserting an item into a random binary tree of size N

    1.15. Inserting an item into a left-leaning red-black tree of size N

    1.16. Insert 1 element into a hash-table that has N elements.

    1.17. Find an item in a hash-table that has N items.

1.18. For a graph G = (V,E) with no negative edge weights, how long does Dijkstra's algorithm take to find the shortest path between distinguished vertices s and t.

1.19. How long does the LU factorization take for a matrix of size NxN?

1.20. How long does Strassen's algorithm take to multiply 2 square matrices of size N x N?

2. What is the average case complexity for each of these algorithms

2.1. Selection Sort of N items ?

2.2. Insertion Sort of N items ?

2.3. Shell sort of N items?

2.4. A sequence of M union-find operations on N disjoint sets

2.5. The copying cost when using the doubling algorithm for a sequence of N push operations

2.6. Quicksort of N items

2.7. Merge sort of N items

2.8. Insert 1 item into a priority queue of size N

2.9. A LSB radix sort over an alphabet of size A, for fixed-length strings of length L. there are N such strings

2.10. Finding an item in a random binary search tree of size N

2.11. Finding an item in a left-leaning red-black tree of size N

2.12. Inserting an item into a random binary tree of size N

2.13. Inserting an item into a left-leaning red-black tree of size N

2.14. Insert 1 element into a hash-table that has N elements.

2.15. Find an item in a hash-table that has N items

2.16. For a graph G = (V,E) with no negative edge weights, how long does Dijkstra's algorithm take to find the shortest path between distinguished vertices s and t.

2.17. Given a graph G with no negative cycles, but might have negative edge weights, what is the complexity of Dijkstra's algorithm to find the shortest path between 2 distinguished vertices?

2.18. Given a graph G = (V,E) with negative edge weights, but no negative cycles, what is the best known asymptotic complexity for finding the shortest path between 2 distinguished vertices s and t?

2.19. Given a graph G = (V,E) with negative edge weights, but no negative cycles, what is the best known asymptotic complexity for finding the shortest path from 1 distinguished vertex s to all other vertices in the graph?

2.20. Given a graph G = (V,E), with weighted edges given by the function w(e) for $e \in E$, what is the best known asymptotic complexity to find the minimum cost spanning tree?

2.21. How long does the LU factorization take for a matrix of size NxN?

2.22. How long does Strassen's algorithm take to multiply 2 square matrices of size N x N ?

Not-so-short Answer Questions

1. What is the computational complexity of the following program? (Big Oh notation is sufficient). Assume that `X[i,j]` references the `(i,j)` component of matrix `X`. Also, assume that `N` parameter describing the size of `X` as `X[N,N]`.

```
for (i=1;i<=N;i++){
     v[i] = 0;
     for (j=i+1;j<=N;j++) {
          v[i] += A[i,j]*b[j];
     }
 }
```

   Be sure to give your reasons for your complexity calculation

2. For the Union-Find f array below:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
f[i]	2	2	4	2	6	7	2	4	5	6	11	13	11	16	11	11

   2.1. Draw the forest corresponding to the f-array.

   2.2. Can the above be a result of running weighted quick union? Explain why this is impossible OR ELSE give a sequence of union-find operations that produce the above table.

3. Using any method you like, find the Big Oh complexity of the recurrence

   $T(n) = 5T(n/4) + n^3 \log(n)$

4. Using any method you like, find the Big Oh complexity of the recurrence

   $T(n) = T(2n/3) + T(n/3) + n$

5. You are building a web crawler and storing each full HTML URL in an array. You notice this taking up a lot of room on your laptop. You also notice that the HTML URLs have a lot of common prefixes.

   What data structure do you recommend to avoid storing more that 1 copy of each prefix?

   How would you find the longest common prefix (LCP) of 2 URLs in your data structure?

   What is the complexity of your LCP algorithm?

6. Show a trace of quickselect to find the median of the following sequence (use 1st element pivots):

```
3, 17, -5, 4, 13, 8, 7, 6, 9, 15,-15
```

7. A variation of the rod-cutting problem is the following:

   Given a rod of length n inches, a table of prices p_i for i = 1, 2 ..., and a cost C for cutting a rod, determine the maximum revenue R obtainable by cutting up the rod and selling the pieces.

   Example:

   Consider the case when n = 6, with the price (in $) breakdown given below, and the cost of a cut is $1.

length i	1	2	3	4	5	6	7	8	9	10
price p_i	1	5	8	9	10	17	17	20	24	30

   One possible way choice is not to cut the rod at all, but to sell it intact for $9

   Another alternative would be to cut it into 6 1 inch pieces for $6 in hardware sales - $5 in cutting cost, giving a net profit of $1 — Not as good as our first option.

   7.1. Give an algorithm that solves the rod cutting problem for any length rod and price breakdown list.

   7.2. Show a trace of your algorithm running to solve the initial example shown above

8. You are given a list of meeting times as pairs of numbers. For example:

    #1 (7, 9) = from 7 AM to 9 AM
    #2 (15, 16) = from 3 PM to 4 PM
    #3. (8, 11) = from 8 AM to 11 AM

   In the list, some meetings will overlap. For example meeting #1 and meeting #3 overlap.
   For this problem, you should find an algorithm to merge all overlapping meetings.
   Your output should be a list of new meeting times that do not overlap.

   In the given example, your final output would be:

    #1 (7, 11) = 7 AM to 11 AM, merge #1 & #3 above
    #2 (15, 16). = 3 PM to 4 PM (nothing to merge)

   Note that consecutive meetings should be merged. (2,3) and (3,4) should be merged into (2,4)

   Similarly, meetings that are subsumed should be merged into the inclusive meeting. (1, 5) and (2,4) should be merged into (1,5).

   The input to your merging algorithm will be an array of N pairs. Give the complexity of your algorithm in terms of N.

9. Given the following set of strings:

   abe
   baa
   eee
   abc
   bad
   bab
   abc

   Every string has length 3 (ignore leading spaces). Furthermore, every string is drawn from only 5 characters: a,b,c,d,e
   What algorithm has the best asymptotic complexity to sort this set?
   Write pseudocode for that algorithm, and show a trace of the sorting procedure.

10. Solve the following linear system using the LU factorization and triangular solves
    Be sure to show your steps, including factorization
    Answers can be given in decimals if you want.

$$\begin{pmatrix} 1.0 & 2.0 & 3.0 \\ 2.0 & 3.0 & 4.0 \\ 3.0 & 4.0 & 5.0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 10 \\ 12 \\ 15 \end{pmatrix}$$

11. A cycle in a directed graph. G = (V,E) is defined as a path $<v_1,v_2,\ldots v_1>$, where the last vertex is the same as the first vertex.Given a directed graph G=(V,E), determine if G has a directed cycle.
    Hint: Use Depth First Search

12. You are given 2 strings $X=x_1x_2x_3x_4\ldots x_m$ and $Y=y_1y_2y_3\ldots y_n$. The **weighted edit distance** is defined as the minimum cost of an edit instruction sequence to convert string X to string Y.
The edit operations are:
   1. d: delete a character, cost = c_d
   2. i: insert a character, cost = c_i
   3. r: replace a character, cost = c_r

   Note that these operations, however, do not necessarily have the same cost. For example, cost(d) = cost(i) = 1, and cost(r) = 4. then sequences X = a and Y = b, then edit distance = 2 —> delete(a),insert(b). The alternative replace(a,b) has cost(r) = 4. For cost(r) = cost(i) = cost(d) = 1 then weighted edit distance for X, Y = replace(a,b), so edit distance = 1.

   Devise an efficient algorithm to solve the weighted edit distance problem.

13. Let G = (V,E) be a graph with edge weights given by w(E). Suppose we also know that that

   $$1 \leq w(E) \leq 10.$$

   Give a modification to Kruskal's algorithm for the minimum spanning tree that exploits the bounds on the edge weights. What is asymptotic complexity of your modified algorithm?