Computer Science 477/677 Fall 2021

Practice for the Final Examination December 8, 2021

The entire practice test is 695 points.

1. True or False. Write "O" if the answer is not known to science at this time. [5 points each]

   (a) _____ There is a polynomial time algorithm for finding the factors of any integer written as a base ten numeral.

   (b) _____ Planar graphs are sparse.

   (c) _____ Tree graphs are sparse.

   (d) _____ Acyclic directed graphs are sparse.

2. Fill in the blanks.

   (a) [5 points] If a planar graph $\mathcal{G}$ has 5 vertices, then the number of edges of $\mathcal{G}$ cannot be more than _____. (You must give the best possible answer, exactly. No partial credit.)

   (b) [5 points] The height of a binary tree with 10 nodes is at least _____. (You must give the best possible answer, exactly. No partial credit.)

   (c) [5 points] A directed graph has a topological order if and only if it is _____.

   (d) [15 points] _____ _____ and _____ are three examples of priority queues.

   (e) [5 points] The operators of the ADT _____ are *pop* and *push*.

   (f) [5 points] The operators of the ADT _____ are *fetch* and *store*.

   (g) [10 points] In order to solve a shortest path problem on a weighted directed graph, there must be no _____ _____. (2 words)

   (h) [10 points] The following is pseudo-code for what algorithm?

   _____

```
int x[n];
obtain values of x;
for(int i = 0; i < n; i++)
 for(int j = i+1; j < n; j++)
  if(x[i] > x[j])
   swap(x[i],x[j]);
```

   (i) [10 points] The asymptotic time complexity of Johnson's algorithm on a weighted directed graph of n vertices and m arcs is _____. (Your answer should use O notation.)

   (j) [5 points] The time complexity of every comparison-based sorting algorithm is _____ . (Your answer should use Ω notation.)

   (k) [10 points] The postfix expression $xy+xz-*$ is equivalent to the infix expression _____.

(l) [10 points] The items stored in a priority queue (that includes stacks, queues, and heaps) represent _____ _____. (2 words)

(m) [10 points] Name two divide-and-conquer sorting algorithms.

_____

_____

3. Give the asymptotic complexity, in terms of `n`, of each of the following code fragments. [10 points each]

(a)
```
for(i = 0; i < n; i = i+1);
cout << "Hi!" << endl;
```

(b)
```
for(int i = n; i > 1; i = i/2)
  cout << "hello world" << endl;
```

(c)
```
for(int i = 1; i < n; i++)
  for(int j = 1; j < i; j = 2*j)
    cout << "hello world" << endl;
```

(d)
```
for(int i = 1; i < n; i++)
  for(int j = i; j < n; j = 2*j)
    cout << "hello world" << endl;
```

(e)
```
for(int i = 2; i < n; i = i*i)
  cout << "hello world" << endl;
```

(f)
```
for(i = 1; i < n; i = 2*i);
cout << "Hi!" << endl;
```

(g)
```
for(int i = n; i > 1; i = sqrt(i));
 cout << "Hi!" << endl;
```

4. For each of these recursive functions, write, and then solve, an appropriate recurrence. [10 points each]

(a)
```
int f(int n)
{
 if (n < 2) return 1;
 else return f(n-1)+f(n-1);
}
```

(b)
```
void g(int n)
 {
  if(n >= 1)
   {
    for(int i = 1; i < n; i++)
     cout << "Hi!" << endl;
    g(n/3);
    g(n/3);
    g(n/3);
   }
 }
```

5. Solve the recurrences. Give the asymptotic value of $F(n)$ using either O, Ω, or Θ, whichever is most appropriate. [10 points each]

(a) $F(n) = 2F\left(\frac{n}{2}\right) + n$

(b) $F(n) \geq 4F\left(\frac{n}{2}\right) + n^2$

(c) $F(n) = F(n-1) + \frac{n}{4}$

(d) $F(n) \leq F\left(\frac{n}{2}\right) + F\left(\frac{n}{4}\right) + F\left(\frac{n}{5}\right) + n$

(e) $F(n) = F\left(n - \sqrt{n}\right) + \sqrt{n}$

(f) $F(n) = F(\log n) + 1$

(g) $T(n) < T(n-2) + n^2$

(h) $F(n) \geq F(\sqrt{n}) + \lg n$

(i) $G(n) \geq G(n-1) + n$

(j) $F(n) = 4F(n/2) + n^2$.

(k) $F(n) \leq 2F(\sqrt{n}) + O(\log n)$.

(l) $K(n) = K(n - \sqrt{n}) + 1$.

(m) $F(n) = 4F\left(\frac{3n}{4}\right) + n^5$ (No, you don't need a calculator.)

(n) $F(n) = F(n/2) + 1$

(o) $F(n) = F(n-1) + O(\log n)$

(p) $F(n) = F\left(\frac{n}{2}\right) + 2F\left(\frac{n}{4}\right) + n$

(q) $F(n) = F\left(\frac{3n}{5}\right) + F\left(\frac{4n}{5}\right) + n^2$

(r) $F(n) = F(n-2) + n$

6. [20 points] Use dynamic programming to compute the length of the longest common subsequence of the strings "011011001" and "1010011001." Show the matrix.

7. [20 points] Use dynamic programming to compute the Levenshtein distance between the strings "kitchen" and "chicken." Show the matrix.

8. [20 points] Describe a dynamic programming algorithm to compute the maximum sum of any contiguous subsequence of a given sequence of numbers. For example, if the given sequence is

$$2, 1, -4, 6, -3, 7, -1.2. - 2, 1$$

that sum is $6 + (-3) + 7 + (-1) + 2 = 11$. (There is a $\Theta(n)$-time algorithm.)

9. Use Huffman's algorithm to construct an optimal prefix code for the alphabet $\{A, B, C, D, E, F\}$ where the frequencies of the symbols are given by the following table.

A	2
B	8
C	9
D	3
E	7
F	5

10. [10 points] Write pseudo-code for binary search.

11. [20 points] What does the following recursive function do? Hint: it is important for cryptography.

```
int f(int m, int n, int e)
 {
  assert(m > 0 and n > 0 and e >= 0);
  if(e == 0) return 1;
  else if(e%2) return f(m,n,e-1)*m%n;
  else
   {
    temp = f(m,n,e/2);
    return temp*temp%n;
   }
 }
```

12. [20 points] The usual recurrence for Fibonacci numbers is:
$F[1] = F[2] = 1$
$F[n] = F[n-1]$ for $n > 2$

However, there is more efficient recurrence:
$F[1] = F[2] = 1$
$F[n] = F\left[\frac{n-1}{2}\right] * F\left[\frac{n}{2}\right] + F\left[\frac{n+1}{2}\right] * F\left[\frac{n+2}{2}\right]$ for $n > 2$
where integer division is truncated as in C++.

Using that recurrence, Describe a $\Theta(\log n)$-time memoization algorithm which reads a value of n and computes $F[n]$, but computes only $O(\log n)$ intermediate values.

13. [20 points] Describe a randomized algorithm which finds the k^{th} smallest element of an unsorted list of n distinct numbers, for a given $k \leq n$, in $O(n)$ expected time. (By "distinct," I mean that no two numbers in the list are equal.)

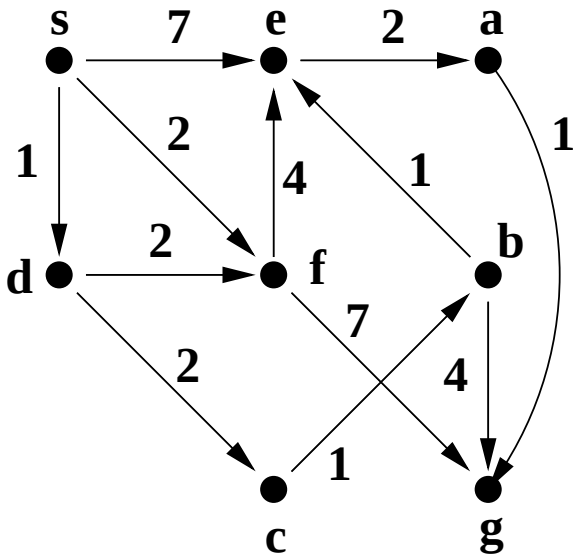
14. [20 points] Give pseudocode for the Bellman-Ford algorithm.

15. [20 points] Give pseudocode for the Floyd-Warshall algorithm.

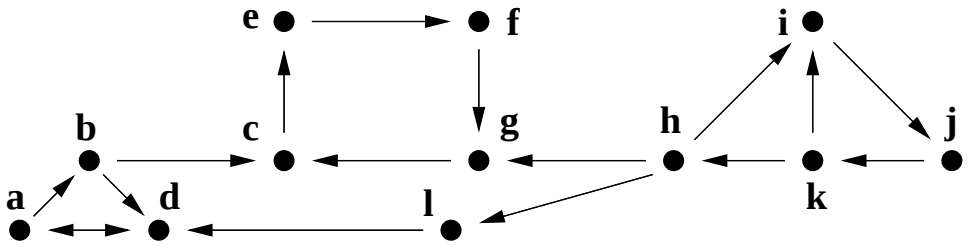
16. [20 points] Describe, in a very informal way, an $O(\log n)$ time algorithm which computes the median of two sorted lists of length n.

17. [20 points] State Hall's Marriage Theorem.

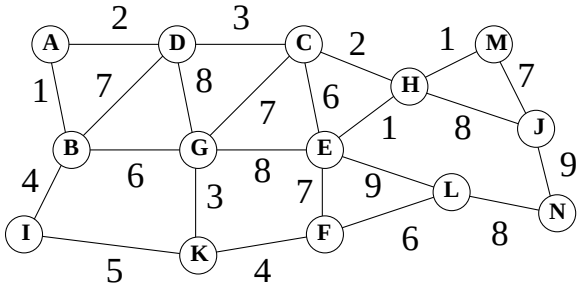
18. [20 points] Use Dijkstra's algorithm to solve the single source shortest path problem for the following weighted directed graph, where **s** is the source. Show the steps.



19. [20 points] Find the strong components of the following graph, using DFS search. (If you find a better algorithm than the one I showed you in class, you may use that instead.) Circle the strong components.



20. [20 points] Show the minimum spanning tree of the following weighted graph. Show the evolution of the union-find structure.



21. Consider the function george computed by the following recursive function.

```
int george(int n)
 {
  if(n <= 6) return 1;
  else return george(n/2)+george(n/2+1)+george(n/2+2)+george(n/2+3)+n*n;
 }
```

(a) [10 points] What is the asymptotic complexity of george(n), in terms of n? Use Θ notation.

(b) [10 points] What is the asymptotic time complexity of the recursive code, in terms of n? Use Θ notation.

(c) [10 points] What is the asymptotic time complexity, in terms of n, of a dynamic program algorithm which computes george(i) for all i up to n? Use Θ notation.

(d) [20 points] What is the asymptotic time complexity, in terms of n, of a memoization algorithm which computes george(n)? Use Θ notation.