Study for CS477 Final Examination May 6, 2024

1. In each blank, write Θ if correct, otherwise write O or Ω, whichever is correct.

   (i) $n^2 =$ _____ (n^3)

   (ii) $\log(n^2) =$ _____ $(\log(n^3))$

   (iii) $\log(n!) =$ _____ $(n \log n)$

   (iv) $\log_2 n =$ _____ $(\log_4 n)$

   (v) $n^{0.000000000001} =$ _____ $(\log n)$

   (vi) $\log^* \log n =$ _____ $(\log^* n)$

2. True or False. Write "O" if the answer is not known to science at this time.

   (i) _____ No good programmar would ever implement a search structure as an unordered list.

   (ii) _____ Computers are so fast nowadays that there is no longer any point to analyzing the time complexity of a program.

   (iii) _____ A complete graph of order 4 is planar.

   (iv) _____ There is a mathematical statement which is true, yet cannot be proven.

   (v) _____ The subproblems of a dynamic program form a directed acyclic graph.

   (vi) _____ Kruskal's algorithm uses dynamic programming.

   (vii) _____ A hash function should appear to be random, but cannot actually be random.

   (viii) _____ Open hashing uses open addressing.

   (ix) _____ Heapsort can be considered to be a sophisticated implementation of selection sort.

   (x) _____ Binary tree sort (also called "treesort") can be considered to be a sophisticated implementation of insertion sort.

3. Fill in the blanks.

   (i) If a planar graph has 7 edges, it must have at least _____ vertices. (You must give the best possible answer, exactly. No partial credit.)

   (ii) The height of a binary tree with 17 nodes is at least _____. (You must give the best possible answer, exactly. No partial credit.)

   (iii) The following is pseudo-code for what algorithm? _____

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

(iv) _____ algorithm does not allow the weight of any arc to be negative.

(v) 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.)

(vi) The time complexity of every comparison-based sorting algorithm is _____ . (Your answer should use Ω notation.)

(vii) The postfix expression $zw + x \sim y - *$ is equivalent to the infix expression _____.

(viii) The items stored in a priority queue (that includes stacks, queues, and heaps) represent _____ _____.

(ix) The asymptotic complexity of the Floyd/Warshall algorithm is _____.

(x) The asymptotic complexity of Dijkstra's algorithm algorithm is _____.

(xi) A _____ hash function fills the hash table exactly with no collisions.

(xii) _____ algorithm finds a binary code so that the code for one symbol is never a prefix of the code for another symbol.

(xiii) _____ and _____ are greedy algorithms that we've studied this semester.

(xiv) _____ and _____ are divide-and-conquer algorithms that we've studied this semester.

(xv) An acyclic directed graph with 9 vertices must have at least _____ strong components. (Must be exact answer.)

(xvi) In _____ _____ there can be any number of items at a given index of the hash table.

(xvii) The asymptotic expected time to find the median item in an unordered array of size n, using a randomized selection algorithm, is _____.

(xviii) If a directed acyclic graph has n vertices, it must have _____ strong components.

(xix) If a planar graph has 10 edges, it must have at least _____ vertices.

(xx) Fill in this blank with one letter.
If all arc weights are equal, then Dijkstra's algorithm visits the vertices in same order as _____FS.

(xxi) The following is pseudo-code for what algorithm? _____

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

(xxii) The prefix expression $*a+ \sim b * -c\,d \sim e$ is equivalent to the infix expression

_____ and the postfix expression _____.

(xxiii) In _____ hashing, each item has more than one hash value, but only uses one of them.

4. Give the asymptotic complexity, in terms of `n`, of each of the following code fragments.

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

(ii)
```
for(int i = 1; i < n; i = i+i)
  cout << "Hello world" << endl;
```

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

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

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

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

(vii)
```
for(int i = 0; i < n; i++)
  for(int j = n; j > i; j = j/2)
```

(viii)
```
for(int i = 0; i < n; i++)
  for(int j = i; j > 0; j = j/2)
```

(ix)
```
for(int i = 2; i < n; i=i*i)
  cout << "Hello world!" << endl;
```

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

5. Solve the recurrences. Give the asymptotic value of $F(n)$ in terms of n, using Θ notation.

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

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

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

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

(v) $F(n) = 2F(n/2) + n^2$

(vi) $F(n) = 3F(n/9) + 1$

(vii) $F(n) = 4F(n/2) + n^2$

(viii) $F(n) = F(\sqrt{n}) + 1$

(ix) $F(n) = F(3n/5) + 4F(2n/5) + n^2$

(x) $F(n) = F(n/5) + F(7n/10) + n$

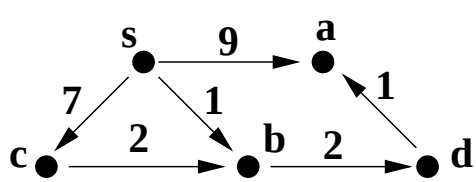
(xi) $F(n) = 2F(n/4) + \sqrt{n}$

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

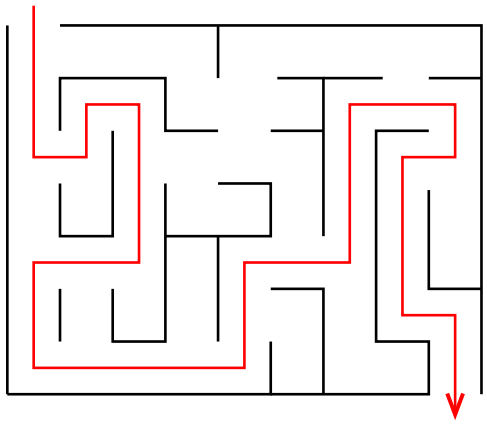
   However, there is another recurrence:
   $F[1] = F[2] = 1$
   $F[n] = F\left[\frac{n-1}{2}\right] * F\left[\frac{n}{2}\right] + F\left[\frac{n+1}{2}\right] * F\left[\frac{n+2}{2}\right]$ for $n > 2$
   where integer division is truncated as in C++.

   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.

7. 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.



8. The figure below shows an example maze. The black lines are walls. You need to find the shortest path, avoiding the walls, from the entrance at the upper left and the exit at the lower right. The red path shows one such path, although it is not the shortest. Describe a program to find the shortest path from the entrance of such a maze, not necessarily this one, to the exit. You do not need to write pseudocode. Your answer should contain the word, "graph," and should state which search method and which data structure(s) you need to use.



9. Find an optimal prefix code for the alphabet $\{a, b, c, d, e, f\}$ where the frequencies are given in the following array.

a	6
b	4
c	2
d	5
e	20
f	1

10. What is the loop invariant of the loop in the following function?

```
float product(float x, int n)
 {
  // assert(n >= 0);
  float z = 0.0;
  float y = x;
  int m = n;
  while(m > 0)
   {
    if(m%2) z = z+y;
    m = m/2;
    y = y+y;
   }
  return z;
 }
```

11. Compute the Levenstein distance between abcdafg and agbccdfc. Show the matrix.

12. You need to store Pascal's triangle in row-major order into a 1-dimensional array P whose indices start at 0. The triangle is infinite, but you will only store $\binom{n}{k}$ for $n < N$. Write a function I such that $P[I(n,k)] = \binom{n}{k}$ for $0 \le k \le n < N$. For example, $I(3,2) = 8$.

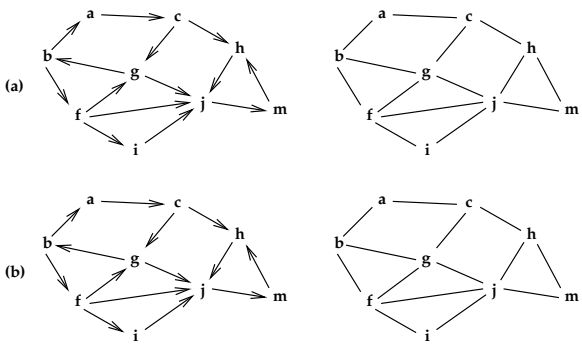
```
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1
```

```
int I(int n, int k)
 {
  // the position of n choose k in the linear array
  assert(k >= 0 and n >= k and n < N);
  int indx =




  return indx;
 }
```

13. Use the DFS method to find the strong components of the digraph shown below as (a). Use the other figures to show your steps.



14. Sketch a circular linked list with dummy node which implements a queue. The queue has four items. From front to rear, these are A, B, C, D, and show the insertion of E into the queue. Show the steps. Don't erase deleted objects; instead, simply cross them out.

HERE

15. You are given an acyclic directed graph $G = (V, E)$ where each arc is weighted. If (x, y) is an arc, we write $w(x, y)$ for the weight of that arc. Describe a dynamic programming algorithm which calculates the directed path through G of maximum weight.

There are two ways to set this problem up. I want you to use the right-to-left method, not the left-to-right. There is one subproblem for each vertex v, namely to compute $M[v]$, the maximum weight of any directed path starting at v. Compute the forward pointer `forw`$[v]$ for each vertex. Explain how those pointers are used to find the path.

16. Write pseudocode for the Bellman-Ford algorithm. Be sure to include the shortcut that ends the program when the final values have been found.

17. List properties of a good hash function.

18. Write pseudocode for the Flyod/Warshall algorithm.

19. What is the loop invariant of the loop in the following function?

```
float product(float x, int n)
 {
  float z = 0.0;
  float y = x;
  int m = n;
  while(m > 0)
   {
    if(m%2) z = z+y;
    m = m/2;
    y = y+y;
   }
  return z;
 }
```

20. Consider the following C++ code.

```
int george(int n)
 {
  if(n == 0) return 1;
  else return george(n/2)+george(n/2-1)+n*n;
 }
```

   (i) What is the asymptotic complexity of `george(n)`?

   (ii) What is the time complexity of the recursive code given above?

   (iii) What is the time complexity of a dynamic programming algorithm to compute `george(n)`?

   (iv) What is the space complexity of a computation of `george(n)` using memoization?

21. Walk through mergesort with the array given below.

    VJATNLDQMEFSPWGL

22. Write pseudcode for the simple coin-row problem we discussed in class. You are given a row of n coins of various values. The problem is to select a set of coins of maximum total value, subject to the condition that no two adjacent coins are selected. Your code should identify the coins which are selected.

23. Write pseudcode for the variation of the coin-row problem where You are given a row of n coins of various values, and you must select a set of coins of maximum total value, subject to the condition that no three adjacent coins are selected. Your code should identify the coins which are selected.

24. Write pseudocode for a function `float power(float x, int n)` that returns x^n. You may assume that $x \neq 0$, but your code must work for any integer n. It is not necessary to use the algorithm given in class; use any $O(\log n)$ time algorithm.

25. The following code is used as a subroutine for both quicksort and select. Assume A[n] is an array of integers. For simplicity, we assume that no two entries of A are equal. Write a loop invariant for the while loop.

```
int pivot = A[0];
int lo = 0;
int hi = n-1;
while(lo < hi)
 {
  if(A[lo+1] < pivot) lo++;
  else if(A[hi] > pivot) hi--;
  else swap(A[lo+1],A[hi]);
 }
```

26. Fill in the blanks.

    (i) _____ is a fast implementation of selection sort.

    (ii) _____ is a fast implementation of insertion sort.

    (iii) The recurrence $F(n) = F(n/5) + F(7n/10) + n$ is used to compute the time complexity of _____.

    (iv) The asymptotic expected height of a treap with n nodes is _____.

    (v) If G is a weighted digraph, it is impossible to solve any shortest path problem on G if G has a _____ _____

    (vi) The height of a binary tree with 45 nodes is at least _____. (You must give the exact answer. No partial credit.)

    (vii) The following is pseudo-code for what algorithm? _____

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

(viii) In closed hashing, if the position at $h(x)$ is already occupied for some data item x, a _____ sequence is used to find an unoccupied position in the hash table.

(ix) A planar graph with $n \geq 3$ vertices can have no more than _____ edges. (Exact formula, please.)

27. What is the loop invariant of the loop in the following function?

```
float product(float x, int n)
 {
  // assert(n >= 0);
  float z = 0.0;
  float y = x;
  int m = n;
  while(m > 0)
   {
    if(m%2) z = z+y;
    m = m/2;
    y = y+y;
   }
  return z;
 }
```

28. The following code is used as a subroutine for both quicksort and select. Assume A[n] is an array of integers. For simplicity, we assume that no two entries of A are equal. Write a loop invariant for the while loop.

```
int pivot = A[0];
int lo = 0;
int hi = n-1;
while(lo < hi)
 {
  if(A[lo+1] < pivot) lo++;
  else if(A[hi] > pivot) hi--;
  else swap(A[lo+1],A[hi]);
 }
```

29. Use dynamic programming to compute the Levenshtein edit distance between the strings "abcdef" and "bedafc." Show the matrix.

30. Write pseudo-code for the Floyd/Warshall algorithm. Let the vertices be $\{1, 2, \ldots n\}$. Let $W(i, j)$ be the given weight of the arc (i, j), if any, where $W(i, j) = \infty$ if there is no arc. Compute $V(i, j)$, the minimum weight of any path from i to j, and $B(i, j)$, the backpointer for that minimum path.

31. Walk through polyphase mergesort with the array given below.

    ACBXFREYGMQSNDZ

32. A compiler stores an array `A[8][10][18]` into main memory in row major order, with base address B, and each entry of A requires one place in main memory. Write a formula for the main memory address of `A[i][j][k]` for integers i, j, and k within range.

33. Consider an array implementation of a stack of integers, as given below. Fill in the code which implements the needed operators of a stack.

```
const int N = // whatever
struct stack
 {
  int item[N];
  int size; // number of items in the stack
  // bottom of the stack is at item[0];
 };
void initialize(s&stack)
 {

 }
void push(s&stack,int i)
 {



 }
 bool empty{s&stack)
  {

  }
 int pop(s&stack)
  {



  }
```

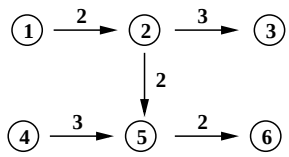
34. In class, we implemented a minheap as an almost complete binary tree implemented as an array. Suppose the minheap is initialized as shown in the first line of the array shown below. Show the evolution of the structure when deletemin is executed.

35. Starting from the configuration given, show the evolution of the structure when B is inserted.

A	C	F	D	Q	H	L	R	Z

C	D	F	R	Q	H	L	Z	

36. You are given an acyclic directed graph $G = (V, E)$ where $V = \{1, 2, \ldots n\}$, and where each arc is weighted. If (i, j) is an arc, we write $W(i, j)$ for the weight of that arc. Give a dynamic programming algorithm which calculates the directed path through G of maximum weight. For example, in the digraph shown below, the maximum weight directed path is (1,2,5,6). Assume that $i < j$ for any arc (i, j), as in the figure. You need not write pseudo-code if you can explain the algorithm without it.



37. Using one of the algorithm we mentioned in class, find the convex hull of the set of points indicated in the figure below. Show your steps.