University of Nevada, Las Vegas Computer Science 477/677 Fall 2022

Assignment 7: Due Friday December 9, 2022, Midnight

This version: Mon Dec 5 08:16:27 PST 2022

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. Please follow Mr. Wang's instructions on how to submit your completed assignment.

1. In each blank, write Θ if correct, otherwise write O or Ω, whichever is correct.
   (a) $n^2 =$ ________ (n^3)
   (b) $\log(n^2) =$ ________ $(\log(n^3))$
   (c) $\log(n!) =$ ________ $(n \log n)$
   (d) $\log_2 n =$ ________ $(\log_4 n)$
   (e) $n^{0.000000000001} =$ ________ $(\log n)$
2. True or False. Write "T" or "F." If the answer is not known to science at this time, write "O" for "Open."
   (a) ________ There is a mathematical statement which is true, yet cannot be proven.
   (b) ________ The subproblems of a dynamic program form a directed acyclic graph.
   (c) ________ A hash function should appear to be random, but cannot actually be random.
   (d) ________ Open hashing uses open addressing.
   (e) ________ No good programmer would ever implement a search structure as an unordered list.
   (f) ________ Computers are so fast nowadays that there is no longer any point to analyzing the time complexity of a program.
   (g) ________ A complete graph of order 4 is planar.
   (h) ________ Heapsort can be considered to be a sophisticated implementation of selection sort.
   (i) ________ Binary tree sort (also called "treesort") can be considered to be a sophisticated implementation of insertion sort.
3. Solve each recurrence, expressing the answer as an asymptotic function of n. Use O, Ω, or Θ, whichever is most appropriate.
   (a) $F(n) \leq 2F(n/2) + n^2$
   (b) $F(n) \geq 3F(n/9) + 1$
   (c) $F(n) = F(3n/5) + 4F(2n/5) + n^2$

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

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

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

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

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

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

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

(k) $G(n) = G(n/2) + 1$

(l) $K(n) \leq 4K(n/2) + n^2$

(m) $J(n) \geq J(3n/5) + J(4n/5) + 1$

(n) $L(n) = L(n - \sqrt{n}) + n$

(o) $H(n) \leq H(\sqrt{n}) + 1$

4. Give the asymptotic time complexity, in terms of n, for each of these code fragments.

(i)
```
for(int i = 0; i < n; i++)
    for(int j = n; j > i; j = j/2)
     cout << "Hello world!" << endl;
```

(ii)
```
for(int i = 0; i < n; i++)
    for(int j = i; j > 0; j = j/2)
      cout << "Hello world!" << endl;
```

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

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

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

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

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

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

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

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

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

(xii)
```
for(int i = 0; i < n; i++)
  for(int j = 0; j*j < n; j++)
    cout << "Hello world" << endl;
```

(xiii)
```
for(int i = n; i > 1; i = i/2)
  for(int j = 0; j < i; j++)
    cout << "Hello world" << endl;
```

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

5. 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

6. Fill in the blanks.

In problems (i) and (ii), let n be the number of vertices, m the number of arcs, and p the maximum number of arcs in the shortest path between any two vertices.

(i) The asymptotic complexity of the Floyd/Warshall algorithm is ______________________________.

(ii) The asymptotic complexity of Dijkstra's algorithm algorithm is ______________________________.

(iii) A ______________________________ hash function fills the hash table exactly with no collisions.

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

(v) ______________________________ and ______________________________ are greedy algorithms that we've studied this semester.

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

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

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

(ix) If $h(x)$ is already occupied for some data item x, a ______________________ ______________________ is used to find an unoccupied position in the hash table.

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

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

(xii) If G is a weighted graph, then it is impossible to solve the all pairs shortest path problem for G if G has a ______________________________ ______________________________

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

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

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

(xvi) 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]);
```

(xvii) ______________ algorithm does not allow the weight of any arc to be negative.

(xviii) 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.)

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

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

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

(xxii) A directed graph has a topological order if and only if it is ______________.

(xxiii) ______________ ______________ and ______________ are three examples of priority queues.

(xxiv) The operators of the ADT ______________ are *pop* and *push*.

(xxv) The operators of the ADT ______________ are *fetch* and *store*.

(xxvi) In order to solve a shortest path problem on a weighted directed graph, there must be no ______________ ______________.

(xxvii) If a planar graph G has n vertices, where n is at least 3, then G can have no more than ______________ edges. (Exact formula, please.)

7. Compute the Levenshtein distance between *abcdafg* and *agbccdfc*. Show the matrix.

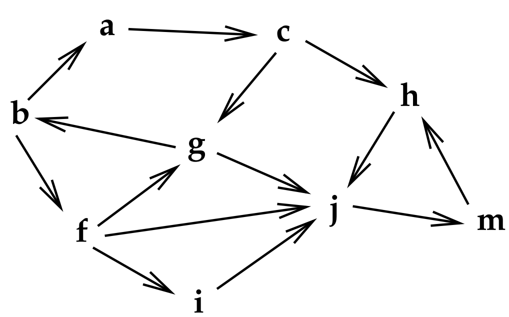
8. You need to store Pascal's triangle in row-major order into a 1-dimensional array P whose indices start at 0. The triangle is infinite, but you will only store $\binom{n}{k}$ for $n < N$. Write a function I such that $P[I(n,k)] = \binom{n}{k}$ for $0 \le k \le n < N$. For example, $I(3,2) = 8$.

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

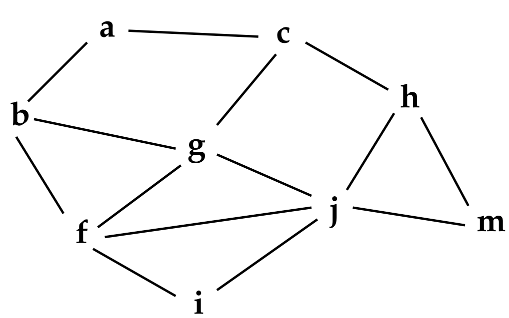
```
int I(int n, int k)
 {
  // the position of n choose k in the linear array
  assert(k >= 0 and n >= k and n < N);
  int indx = // fill in formula here


  return indx;
 }
```

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

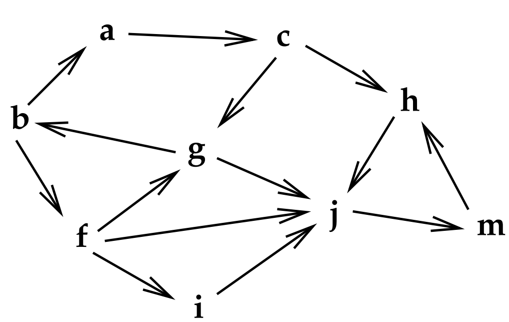
(a)



(b)



(c)



10. 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.

11. In class, we implemented a minheap as an almost complete binary tree, implemented as an array.

(a) 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.

A	C	F	D	Q	H	L	R	Z

(b) Starting from the final configuration above, show the evolution of the structure when B is inserted.

12. 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;
 }
```

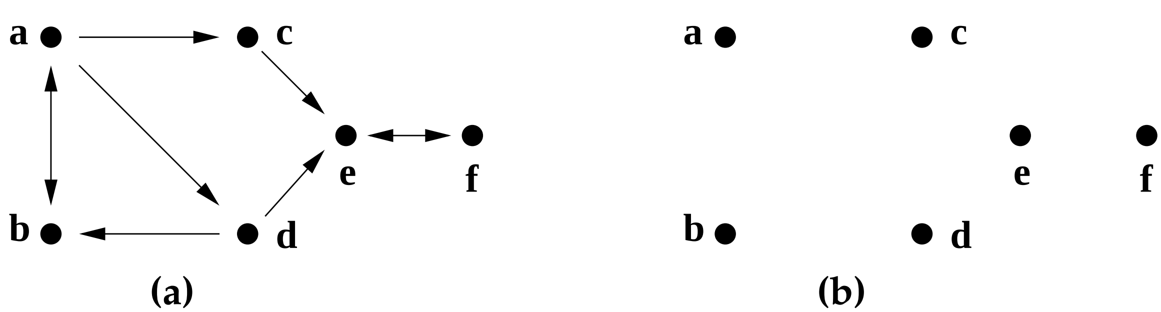
13. The usual recurrence for Fibonacci numbers is:
$F[1] = F[2] = 1$
$F[n] = F[n-1]$ for $n > 2$

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++.

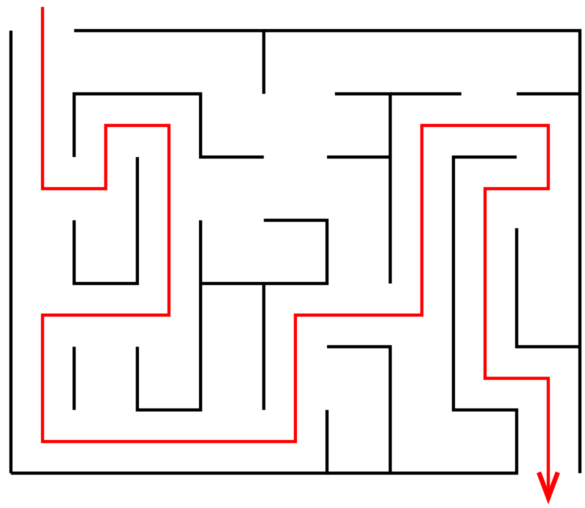
Using that recurrence, Describe a $\Theta(\log n)$-time memoization algorithm which reads a value of n and computes $F[n]$, but computes only $O(\log n)$ intermediate values.

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

15. Find the strong components of graph (a) below, using DFS search. Use (b) for your work. Circle the strong components.



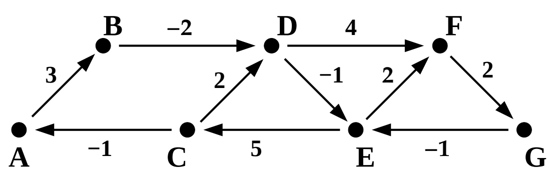
16. The figure below shows an example maze. The black lines are walls. You need to find the shortest path, avoiding the walls, from the entrance at the upper left and the exit at the lower right. The red path shows one such path, although it is not the shortest. Describe a program to find the shortest 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.



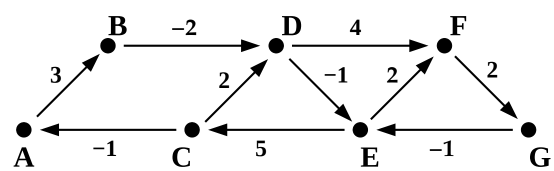
17. Walk through polyphase mergesort, where the input file is as given below.

`VJANLDQMFSP`

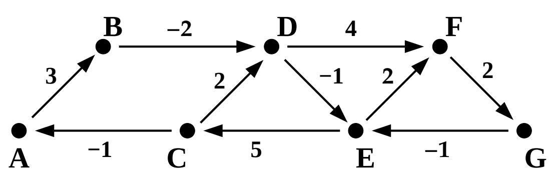
18. Suppose we wish to solve the all-pairs shortest path problem for the weighted digraph shown in (a) below, using Johnson's algorithm. First solve the single source minpath problem using a fictitious source to give each vertex a non-positive number. In (b), label each vertex with that number. We then compute an adusted non-negative weight for each edge. In (c), label each edge with the correct adjusted weight. Do not finish Johnson's algorithm.



(a)



(b)



(c)

19.
```
float power(float x, int n)
{
 assert(n >= 0);
 float y = x;
 int m = n;
 float rslt = 1.0;
 while(m > 0)
  {
   if(m%2) rslt = rslt * y;
   y = y*y;
   n = n/2;
  }
 return rslt;
}
```

(a) What does this function do?

(b) What is the loop invariant of the while loop?