University of Nevada, Las Vegas Computer Science 477/677 Spring 2025

Study for Final Examination May 12, 2025

1. omegatheta1 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) _____ There is a mathematical statement which is true, yet cannot be proven.

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

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

   (v) _____ Heapsort can be considered to be an efficient implementation of selection sort.

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

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

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

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

   (x) _____ Open hashing uses probe sequences.

   (xi) _____ You can avoid collisions in a hash table by making the table twice as large as the data set.

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

   (i)
   ```
   for(i = 0; i < n; i = i+1)
   ```

   (ii)
   ```
   for(int i = 0; i < n; i++)
    for(int j = i; j < n; j = j*j)
   ```

   (iii)
   ```
   for(int i = 1; i < n; i++)
    for(int j = i; j j n; j = 2*j)
   ```

   (iv)
   ```
   for(int i = n; i > 1; i--)
    for(int j = 1; j < i; j = 2*j)
   ```

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

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

(vii)
```
for(int i = 1; i*i < n; i++)
```

(viii)
```
for(int i = 1; i < n; i++)
    for(int j = 1; j < i; j = 2*j)
```

(ix)
```
for(int i = 1; i < n; i++)
    for(int j = 1; j < i; j = 2*j)
```

(x)
```
for(int i = 1; i < n; i++)
    for(int j = 1; j < i; j = 2*j)
```

(xi)
```
for(int i = 1; i < n; i++)
    for(int j = 2; j < i; j=j*j)
```

(xii)
```
for(int i = 1; i < n; i++)
    for(int j = i; j < n; j = 2*j)
```

(xiii)
```
for(int i = 1; i < n; i = i+i)
```

(xiv)
```
for(int i = 2; i < n; i = i*i)
```

(xv)
```
for(int i = 2; i < n; i = i*i)
```

(xvi)
```
for(int i = n; i > 2; i = sqrt(i))
```

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

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

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

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

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

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

(vi) $F(n) = 3F(n-1) + 1$

(vii) $F(n) = 3F(n/2) + n^2$

(viii) $F(n) = 3F(n/3) + 3F(2n/3) + n^2$

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

(x) $F(n) = 4F(n/2) + n$

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

(xii) $F(n) = 4F(n/2) + n$

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

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

(xv) $F(n) = F(n-1) + n^2$

(xvi) $F(n) = F(n/2) + 2F(n/4) + n$

(xvii) $F(n) = F(n/2) + F((n-1)/2) + 3n$

(xviii) $F(n) = F(n/3) + F(n/2) + n$

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

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

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

(xxii) $F(n) = F(n/2) + 2F(n/4) + n$

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

5. Fill in the blanks.

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

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

(iii) The asymptotic complexity of the Floyd/Warshall algorithm for a directed graph of size n with m arcs is _____.

(iv) _____ algorithm finds a binary code for a weighted alphabet such that the code for one symbol is never a prefix of the code for another symbol.

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

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

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

(viii) 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]);
```

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

(x) 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.)

(xi)

(xii) 10 The prefix expression $*a+ \sim b*-c\,d \sim e$ is equivalent to the infix expression _____ and the postfix expression _____.

(xiii) 5 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.

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

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

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

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

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

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

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

(xxiii) 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.

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

(xxv) 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]);
```

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

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

(xxviii) A binary tree with 8 nodes cannot have height less than _____. (Exact answer.)

(xxix) _____ is a divide-and-conquer searching algorithms.

(xxx) _____ and _____ are divide-and-conquer sorting algorithms.

(xxxi) _____ hashing, which can be used by a compiler to identify reserved words, does not have collisions.

(xxxii) In an open hash table, there is a _____ structure at each table index.

(xxxiii) A directed graph is defined to be _____ _____ if, given any two vertices x and y, the graph contains a path from x to y.

(xxxiv) In order for there to exist a topological order of the vertices of a digraph, the graph must be _____.

(i) _____ algorithm does not allow the weight of any arc to be negative.
item The items stored in a priority queue (that includes stacks, queues, and heaps) represent _____ _____.

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

(iii) A _____ hash function has no collisions.

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

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

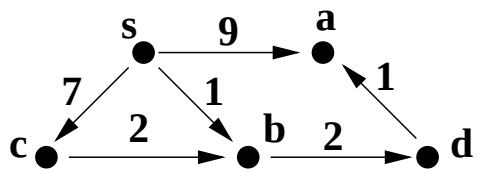
(vi) The asymptotic expected time to find the median item in an unordered array of size n, using a randomized selection algorithm, is _____.

(vii) If a planar graph has 10 edges, it must have at least _____ vertices. (Exact answer. No partial credit.)

(viii) 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]);
```

6. 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.



7. Find an optimal prefix code for the alphabet $\{a, b, c, d, e, f\}$ whose frequencies are given:

a	6
b	4
c	2
d	5
e	9
f	1

8. What is the asymptotic complexity of the function **george**(n), in terms of n?

```
int george(int n)
 {
  // input condition: n >= 0
  if(n < 1) return 1;
```

```
 else return george(n-1)+george(n-1);
}
```

9. What is the asymptotic complexity of the function **martha**(n) in terms of n?

```
int martha(int n)
 {
  // input condition: n >= 1
  if(n == 1) return 0;
  else return n + martha(n/2);
 }
```

10. What is the asymptotic time complexity of above code which computes **martha**(n), in terms of n?

    Hint: This is **not** the same question as the previous one!

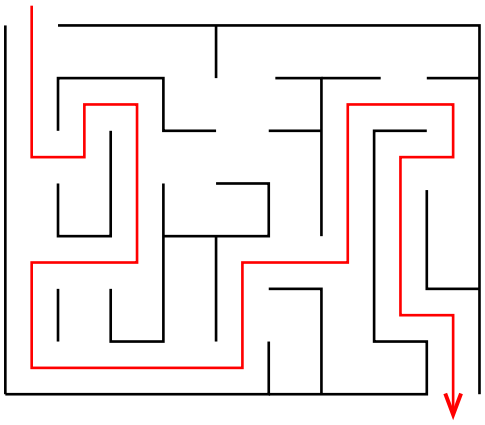
11. Write a C++ function which solves the simplest coinrow problem we have discussed, namely, given a sequence of positive numbers, find the subsequence of maximum total, subject to the condition that the subsequence may not contain any two consecutive terms of the sequence.

12. 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.

13. 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.



14. 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;
 }
```

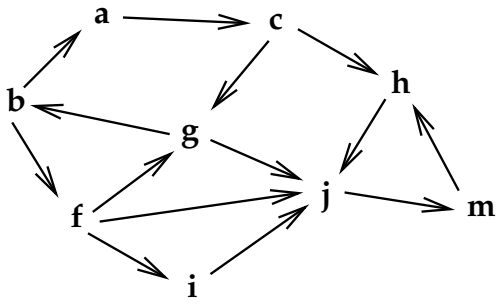
15. A compiler stores an array `A[8][10][18]` into main memory in row major order, with base address B, and each entry of A requires one place in main memory. Write a formula for the main memory address of `A[i][j][k]` for integers i, j, and k within range.

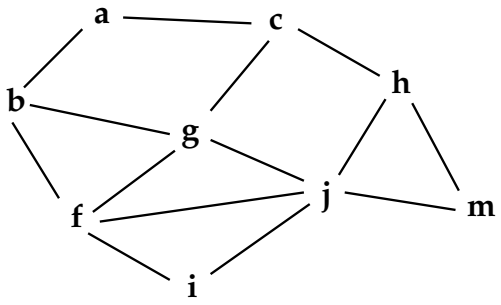
16. A 3-dimensional $8 \times 9 \times 6$ rectangular array X is stored in main memory in column major order, and its base address is 4096. Each item of X takes two words of main memory, that is, two address location. Find the address, in main memory, of X[3][7][4].

17. You are implementing a 3D triangular array A where $A[i][j][k]$ is defined for $0 \le k \le j \le i \le 4$, a total of 35 entries (Is that correct?), and is stored as a one-dimensional subarray of main memory in row-major order, with base address 1024. Each term of A takes one place in main memory. What would be the address, in main memory, of $A[4][2][1]$?

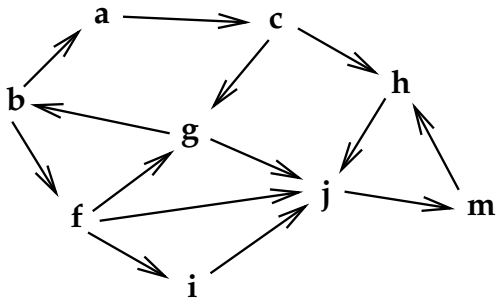
18. Use the DFS method to find the strong components of the digraph shown below as (a). Use the other figures to show your steps.

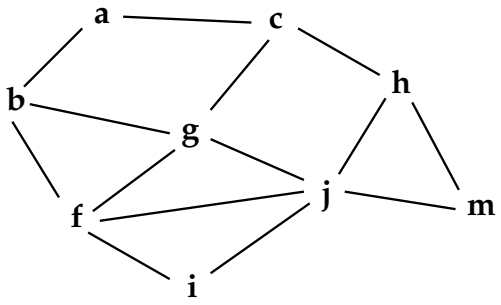
(a)





(b)





19. 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.

    Hint: Subproblem: given a vertex x, what is the maximum weight of any directed path that ends at x?

20. List properties of a good hash function.

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 $x \neq 0$ and $n \geq 0$. It is not necessary to use the algorithm given in class; use any $O(\log n)$ time algorithm.

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

    ACBXFREYGMQSNDZ

26. 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;
}
```

27. 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.

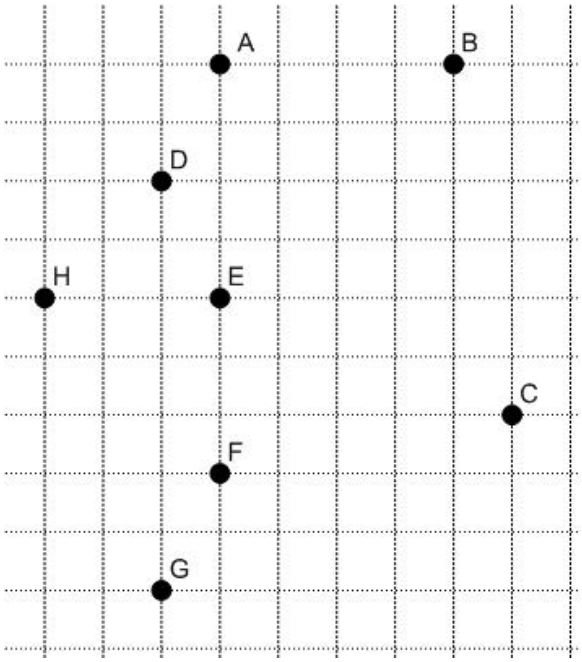
28. Consider an array implementation of a stack of integers, as given below. Fill in the code which implements the needed operators of a stack.

```
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)
  {



  }
```

29. 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 on the left. Show the evolution of the structure when deletemin is executed.

30. Starting from the array given in the first line of the figure on the right, show the evolution of the structure when B is inserted.

31. 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.



32. 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;
 }
```

33. Compute the Levenstein distance between *abcdafg* and *agbccdfc*. Show the matrix.

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

35. 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.

36. 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.

37. Write pseudocode for the Flyod/Warshall algorithm.

38. 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;
 }
```

39. 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.

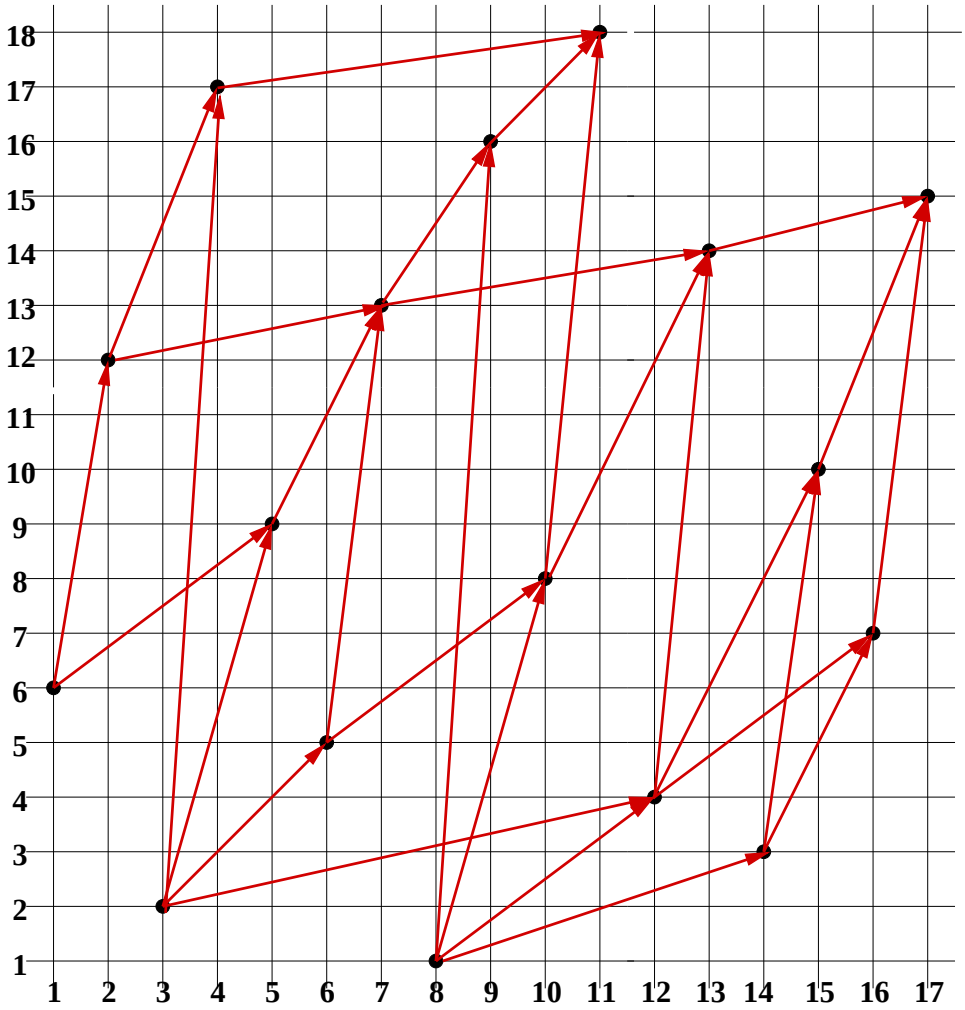
```
int pivot = A[0];
int lo = 0;
int hi = n-1;
while(lo < hi)
 {
  if(A[lo+1] < pivot) lo++;
  else if(A[hi] > pivot) hi--;
  else swap(A[lo+1],A[hi]);
 }
```

40. 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]);
 }
```

41. Find the Longest Monotone Subsequence of the sequence

6,12,1,17,9,5,13,1,6,8,18,4,3,10,7,15



42. omegatheta2 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) $\log_2 n =$ _____ $(\log_4 n)$

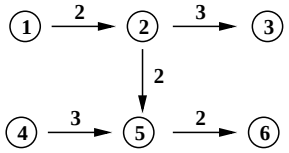
(v) $n^{0.000000000001} =$ _____ $(\log n)$

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

43. 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;
 }
```

44. 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.



45. Write pseudo-code for the Bellman-Ford algorithm. Assume that the the vertices are $\{0, 1, 2, \ldots n\}$ the source is 0, and the arcs are $\{(x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)\}$, and each arc $x_i, y_i)$ has weight W_i. Be sure to include the shortcut that ends the program when the final values have been found.

1. What does the following code compute? What is the loop invariant?

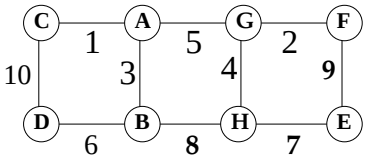
```
float mystery(float x, int n) // input condition: n > 0
 {
  assert(n > 0);
```

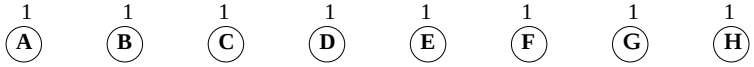
```
  float z = 1.0;
  float y = x*x;
  int m = 2*n;
  while(m > 0)
   {
    if(m%2) z = y*z;
    y = y*y;
    m = m/2;
   }
  return z;
 }
{\tt mystery(x,n)} computes $x^\maly{4n}$.  The loop invariant is
$x^\maly{4n} = y^\maly m*z$.
```

2. Sort the following array using heapsort. Add extra rows if needed.

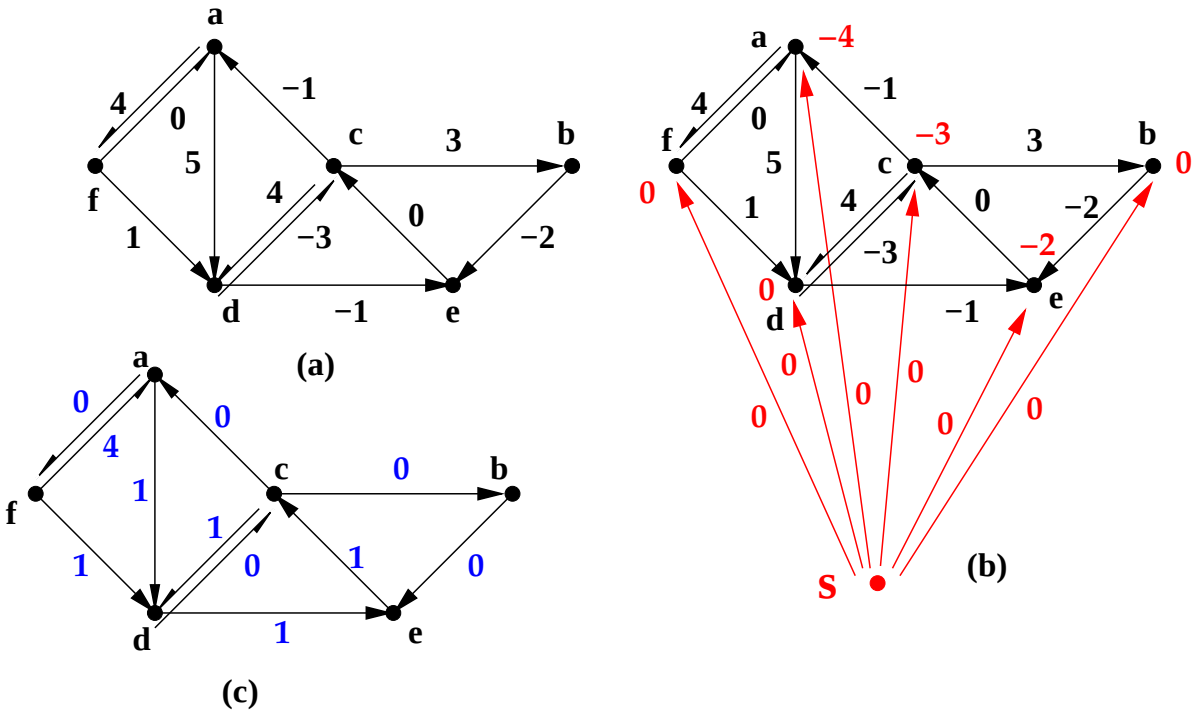
A	N	H	Z	D	V	L	Q

3. Walk through Kruskal's algorithm to find the minimum spanning tree of the weighted graph shown below. Show the evolution of the union/find structure. Whenever there is choice between two edges of equal weight, choose the edge which has the alphabetically largest vertex. Whenever there is a union of two trees of equal weight, choose the alphabetically larger root to be the root of the combined tree. Indicate path compression when it occurs.





4. In the figure below, (a) shows a weighted directed graph. In (c), replace each edge weight using the techniques of Johnson's algorithm. Use (a) and (b) for your work. Do not complete Johnson's algorithm.



5. Find the asymptotic time complexity of each code fragment, in terms of n. Use Θ if possible.

(i)
```
for(int i = 1; i < n; i++)
    for(int j = 1; j < i; j++)
```

(ii)
```
for(int i = 1; i < n; i++)
    for(int j = i; j < n; j++)
```

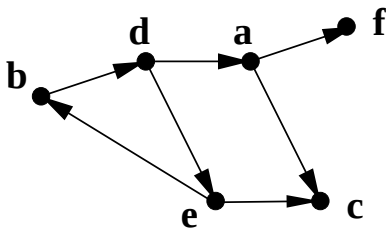
(iii)
```
for(int i = 1; i < n; i = 2*i)
    for(int j = 2; j < i; j = j*j)
```

6. You are trying to construct a cuckoo hash table of size 10, where each of the 10 names listed below has the two possible hash values indicated in the array. Put the items into the table, if possible; otherwise, convince me it's impossible. Instead of erasing ejected items, simply cross them out, so that I can tell that you worked it properly.

	h1	h2
Ann	5	9
Ben	0	8
Cal	4	6
Deb	1	8
Eve	1	0
Fay	9	4
Gus	7	2
Hal	3	7
Ike	3	2
Jan	1	7

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

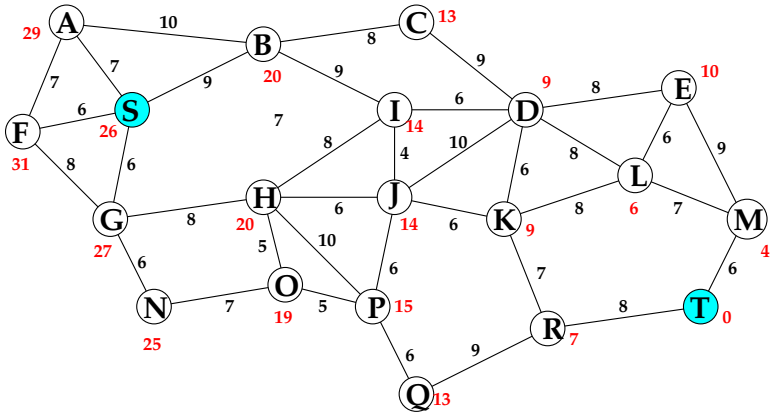
7. Write the in-neighbor list and out-neighbor list representations of the directed graph shown below.



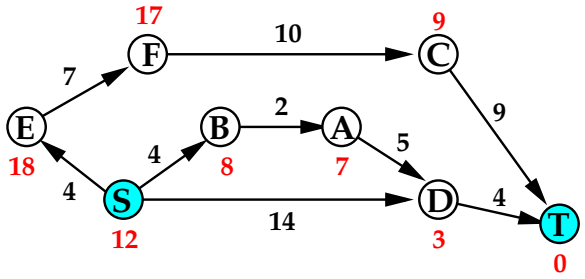
8. Let $\sigma = x_1, x_2, \ldots x_n$ be a sequence of numbers with both positive and negative terms. Write an $O(n)$ time dynamic program which finds the maximum sum of any contiguous subsequence of σ. For example, if the sequence is $-1, 4, -3, 2, 7, -5, 3, 4, -8, +6$ then the answer is $4 - 3 + 2 + 7 - 5 + 3 + 4 = 12$.

9. Explain how to implement a sparse array using a search structure.

10. Walk through the A^* algorithm to find the least cost path from S to T. The values of the heuristic function are indicated in red.



11. Walk through the A^* algorithm for the weighted directed graph shown below, where the pair is (S, T). The heuristic is shown as red numerals.



.

12. Compute the Levenshtein edit distance from the word "proven" to the word "shore." Show the matrix.

13. (a) Find the longest strictly monotone increasing subsequence of the sequence 1,5,2,2,4,8,7. The answer might not be unique. If there are choices, give just one answer.

    (b) Write pseudocode for finding the length of the longest strictly monotone increasing subsequence of any given sequence of integers. (Hint: Use dynamic programming.)

    The algorithm is found in the handout titled, "Longest Monotone Subsequence," which you can find as lms.pdf, under Handouts.

14. Here is another coin-row problem. You have a row of coins of various values, where the value of the i^{th} coin is $V[i] > 0$. Write pseudocode which finds the maximum value of a subset of coins, where the set may not contain coins which are either adjacent or just one apart in the row. That is, if the set contains the i^{th} coin, it may not contain either the $(i+1)^{st}$ coin or the $(i+2)^{nd}$ coin. For example, if the coins are ⓐ ⓑ ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ in that order, the subset may be {ⓐ, ⓓ, ⓗ}, but not {ⓑ, ⓓ, ⓖ}.

15. This is a memoization problem. Values are stored in a sparse array A, whose indices and values are integers. Values can be retrived by executing the function

```
int fetchA(int n)
```

which returns the value of $A[n]$ if it exists, otherwise returns `default`, and the procedure

```
void storeA(int n, int newvalue)
```

Which stores the memo that $A[n] = newvalue$.

To obtain the value $A[n]$ where n is very large, we can execute this function.

```
int A(int n)
 {
  if(n < 2) return 1;
  else
   {
    int temp = fetchA(n);
    if(temp == default)
     {
      int result = A[n/2] + A[n/4] + A[(n-1)/4] + n;
      storeA(n,result);
      return result;
     }
    else return temp;
   }
 }
```

(i) What is the asymptotic value of $A[n]$?

(ii) What is the asymptotic time complexity of the computation of $A[n]$?

(iii) Asymptotically, how many memos are stored in the data structure, when $A[n]$ is computed?