Review Question

1. Which of the following **cannot** be present in a graph if we wish to find the shortest path?
   (a) negative weight edges
   (b) cycles
   (c) cycles with negative weight edge
   (d) negative weight cycles
2. Which of the shortest path algorithm is greedy?
3. Which algorithm would you use on a general graph that has negative weight on edges?
4. Prove that $\forall n \in \mathbb{Z}^+$ then
$$\sum_{i=1}^{n}(-1)^i i^2 = \frac{(-1)^n(n)(n+1)}{2}$$
5. Show that if f(n) and g(n) are monotonically decreasing functions then so are the sum, f(g(n)).
6. A graph with n vertices and n-1 eges is either disconnected or a tree.

   (TRUE/FALSE)
7. For every n there exists a directed graph on n vertices with $\Omega(n^2)$ edge that has a topological ordering.

   (TRUE / FALSE)
8. Solve the recurrence T(n) = $4T(\frac{n}{2}) + O(n)$ by finding its running time.
9. Design the most efficient algorithm you can to combine M sorted list, each of size N into one sorted list. give the time complexity of your solution in terms of N and M.
10. Show that the function f(n) = $n^2 + n \to \Theta(n^2)$
11. Given an array of N non unique values, design the most efficient algorithm you can to find the most common value. Give the time complexity of your solution in terms of N.
12. A minimum weight edge in a connected graph G must belong to every minimum spanning tree for G.

    TRUE/FALSE

13. A minimum weight edge in a connected graph G must belong to <u>some</u> minimum spanning tree for G.

TRUE/FALSE

14. If an edge (u,v) is contained in some minimum spanning tree for graph G then it is a light edge crossing some cut of G.

TRUE/FALSE

15. What problem solving strategy do both Kruskal's and Prim's MST algorithms use?

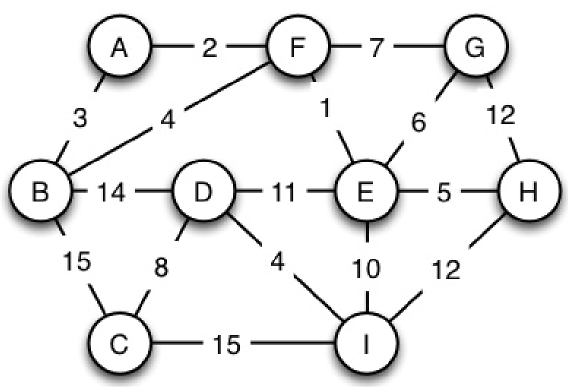
16. Claim: Prim's algorithm correctly finds minimum spanning trees in connected graphs with negative weights.

TRUE/FALSE

17. Claim: Kruskal's algorithm correctly finds minimum spanning trees in connected graphs with negative weights.

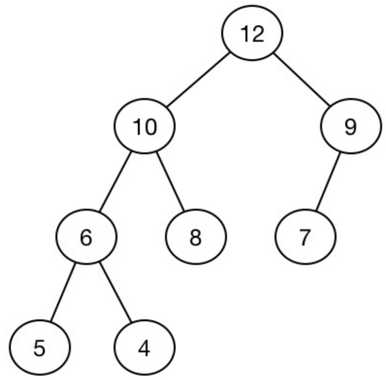
TRUE / FALSE

18. Check off the edges that are included in a Minimum Spanning Tree for this graph.

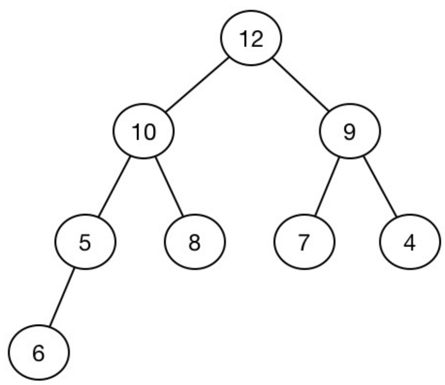


19. In terms of n = $|V|$, Floyd-Warshall's runtime is

20. Is this a Max-Heap? If not, why not?



21. Is this a Max-Heap? If not, why not?



22. Suppose you have a binary heap represented as an array using 1-based indexing (root is at index 1). An item is at index 31 in the array: what is the index of its parent? Write an integer:

23. Suppose you have a binary heap represented as an array using 1-based indexing (root is at index 1). An item is at index 31 in the array: what is the index of its right child? Write an integer:

24. Show the array after Build-Max-Heap has run by entering integers for the keys:

1 2 3 4 5

25. What is the worst-case runtime of randomized quicksort?

26. What is the expected runtime of randomized quicksort?

27. What is the best-case runtime of randomized quicksort?

28. Suppose that you knew that your array was sorted except for the possible misplacement of one or two elements: Which of the sorts that we have studied would be the fastest? What is its expected big-O runtime given your choice in the above question?

29. What is the smallest possible depth of a leaf in a decision tree for a comparison sort of n items?

30. Suppose the deterministic Partition (A, 1, 5) is called on the array A = [7, 3, 9, 4, 5] shown in the table below with 1-based indexing. You will show the state of the array after each swap of elements in the array

31. What is the smallest possible height of a decision tree for a comparison sort of n items?

32. The runtime of counting sort is $\Theta(n + k)$. What is k?

33. The runtime of radix sort using counting sort is $(d(n + k))$. What is d?

34. Which of the sorting algorithms are stable sorts?

35. Use Radix Sort on the following words:

    BOW, DOG, FAX, DIG, BIG, COW

36. Which of the sorting algorithms are not stable sorts?

37. We use Red-Black trees to simultaneously represent what?

38. What of the following properties is violated in a red-black tree when we have underflow of a node in the corresponding 2-4 tree?

39. What of the following properties is necessary to guarantee that the 2-4 tree it represents is balanced, assuming that the other properties hold?

40. What of the following properties is violated in a red-black tree when we have either incorrect representation or overflow of a node in the corresponding 2-4 tree?

41. Match the following with the appropraite sort algorithm

    (a) radix sort
    (b) insertion sort
    (c) merge sort
    (d) Randomized Quick Sort

    (i) application demands a gurantee that every sort finishes within O(n lg n) time even if average cost per higher

(ii) application requires a fast sort on a variety of input sizes and types and distribution of keys and the sort must be done in place to conserve memory

(iii) application will only be very short lists and you want simple code

(iv) you are sorting a set of d digit integers and done mind using a specialized sort to gain speed

42. "Quick Sort can be used as an auxillary sorting routine in Radix Sort, because it operates in places i.e. does not use any extra memory.

TRUE / FALSE

43. Suppose we run DFS on a directed graph using the CLRS version of the algorithm, and while searching vertex v encounter a black node on recursive call. What type of edge is this?

44. Illustrate the operation of PARTITION on the array A $= < 13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11 >$

45. After a given iteration of the loop line 3 of Floyd Warshall, we have computed:

46. Suppose we run DFS on a directed graph using the CLRS version of the algorithm, and while searching vertex v encounter a grey node on recursive call. What type of edge is this?

47. Illustrate a BUILD-MAX-HEAP on the array A $= [5, 3, 17, 10, 84, 19, 6, 22, 9]$

48. What is the **worst case runtime** of Insertion Sort in terms of input size n?

49. If you want a guarantee that an algorithm will never take more than a certain amount of time for a given input size, which analysis do you need?

50. Match the three criteria that a loop invariant must satisfy in order to prove correctness of an algorithm:

    (a) Maintenance

    (b) Termination

    (c) Initialization

    (i) The loop invariant must be true before the first iteration of the loop being analyzed

    (ii) If the loop invariant is true prior to a given iteration of the loop being analyzed, then it remains true before the next iteration

    (iii) If true after the final iteration of the loop being analyzed, the loop invariant states a property that will help show that an algorithm is correct

51. Give the Θ notation for the following expression, simplified as much as possible for $7n^3 + 4n + 99$

52. Consider how Merge Sort implements the Divide and Conquer strategy. At a given call to Merge Sort:
    (a) What is the cost to divide a problem of size n into subproblems?
    (b) What is the cost to combine the solutions to subproblems?
    (c) How many subproblems does Merge Sort divide a problem of size n into?
    (d) What is the size of each subproblem as a function of n?
53. (Ordered DLL) What is the tightest bound on the worst case asymptotic cost of the fastest algorithm for searching an ordered doubly linked list for a given key?
54. What is the worst case and the best case run time for Merge Sort?
55. What is the tightest bound on the worst case asymptotic cost of stack push and pop operations under the array implementations discussed in the text and lecture?
56. What is the tightest bound on the worst case asymptotic cost of queue enqueue and dequeue operations under the array implementations discussed in the text and lecture?
57. (Unordered DLL) What is the tightest bound on the worst case asymptotic cost of the fastest algorithm for searching an unordered doubly linked list for a given key?
58. (SLL) Given a pointer to a list cell in a singly-linked list, what is the expected asymptotic run time of the fastest algorithm to insert a new cell after that cell?
59. (Ordered DLL) What is the tightest bound on the worst case asymptotic cost of the fastest algorithm for insertion of a new key into an ordered doubly linked list? (You have only the key, not the position.)
60. (SLL) Given a pointer to a list cell in a singly-linked list, what is the expected asymptotic run time of the fastest algorithm to delete that cell?
61. (Unordered DLL) What is the tightest bound on the worst case asymptotic cost of the fastest algorithm for insertion of a new key into an unordered doubly linked list?
62. (Binary Tree) Suppose we have a binary tree of n nodes, each node having a key (their ordering is not important for this question), and using a linked node representation (e.g., nodes of class TreeNode as described in the lectures). What is the asymptotic run time of the fastest algorithm to print out the keys of all n nodes of the binary tree?
63. This is a circular doubly linked list with a sentinel referenced by L.nil. What data object does L.nil.prev.prev.key return?

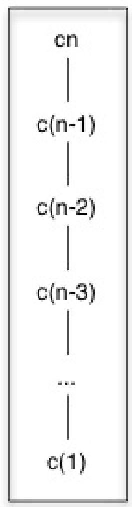
64. Insert the keys 9, 18, and 13 into a hash table using chaining, with m = 5 and h(k) = k mod m. Values to insert (in this order): 9, 18, 13.

65. Solve the following recurrence relation

$$T(n) = 2T(\frac{n}{2}) + n^2$$

66. Insert the keys 9, 18, and 13 into a hash table using open addressing with linear probing, with m = 5 and h(k,i) = (h'(k) + i) mod m. Values to insert (in this order): 9, 18, 13.

67. Write the recurrence relation for the runtime cost of the recursive procedure for which this recursion tree represents the recursion



68. Solve the recurrence relation below

$$T(n) = 4T(\frac{n}{2}) + n^2$$

69. Is the proof correct

Suppose we want to prove that T(n) = 6T(n/3) + n = Θ(n), using proof by substitution. Using c for the constant that is implied in the definition of Θ(n), we guess T(n) ≤ cn, and then prove by substitution of the guess into the formula as follows:

T(n) = 6T(n/3) + n

≤ 6(c(n/3)) + n *// substitution of guess*

≤ 2cn + n *// simplifying*

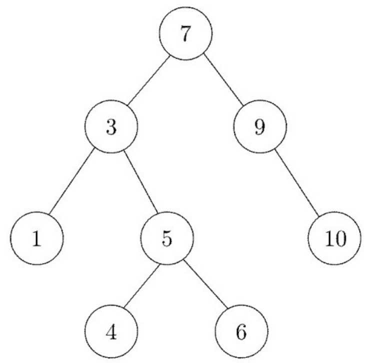
= Θ(n) *// constants don't affect growth rate*

70. How may leaves are there in a complete binary tree of height 10?

71. If G = (V, E) is a connected graph, which of these additional conditions will guarantee that G is a tree?

72. How may internal vertices are there in a complete binary tree of height 10?

73. Is this a valid BST?



74. Which traversal would you use to get the keys of a binary search tree printed out in sorted order

75. If you have an optimal solution to a sub problem, this solution must be part of the optimal solution to the overall problem.

(TRUE / FALSE)

76. Solve the following recurrence relation

(a)
$$T(n) = 2T(\frac{n}{2}) + n^4$$

(b)
$$T(n) = T(n-2) + n^2$$

(c)
$$T(n) = 4T(\frac{n}{2}) + n^2\sqrt{n}$$

77. Suppose that we have an array of n data records to sort and that the key of each record has the value 0 or 1.
An algorithm for sorting such a set of records might possess some subset of the following three desirable characteristics:

    (i) runs in O(n) times
    (ii) stable
    (iii) sorts in place, using no more than a constant amount of storage space in addition to the original array

    Given an algorithm such that it fits

    (a) criteria i and ii
    (b) criteria i and iii
    (c) criteria ii and iii

78. Characterize the running time of the following algorithm with justification

```
TripleLoop(A, n)
  for k = 1 to 10
    for i = 1 to n
      for j = n downto i
        A[i] = A[j] + 1
```

79. How does the growth rate of these pairs of functions compare (O, o, Ω, ω, Θ, or incomparable) and why?

    (a) $n^2 \& 4^{\log(n)}$
    (b) 2^{999} and $lg(lg(n))$
    (c) $lg(n)$ and $\log_{10}(n)$

80. Use induction to prove $d! > 2d$ for all large integers d

81. Given an adjacency list representation of a graph G=(V,E), and assuming randomly distributed graphs, what is the time complexity in terms of V and E to determine whether an edge exists from vertex u to vertex v, and why?

82. Given an array representation A of a graph G=(V,E), and assuming randomly distributed graphs, what is the time complexity in terms of V and E to determine whether an edge exists from vertex u to vertex v, and why?

83. Given a hash table with chaining, having m cells and n items have been inserted into the hash table, what is the expected time in terms of n and m to find an element x in the hash table, and why?

84. The value $|f|$ of a flow f : VxV$\rightarrow \mathbb{R}$ with source s and sink t is:

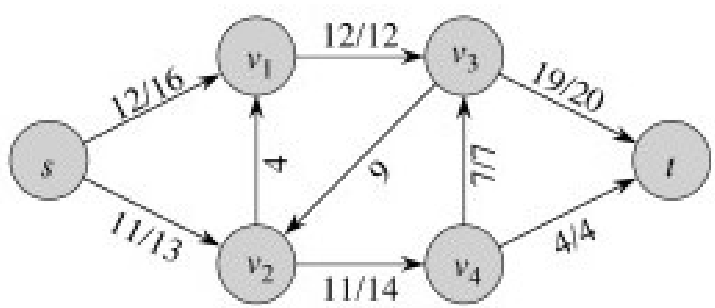
85. Given an n x n array A, what is the Θ time complexity in terms of n to access a cell A[i,j] of the array, and why?

86. Explain the type of the edges:

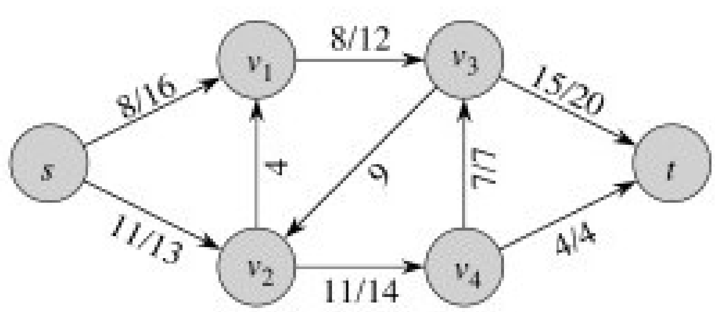
    (a) Forward Edges
    (b) Backward Edges
    (c) Cross Edges
    (d) Tree

87. In a flow network, the flow f(u, v) assigned to an edge has what kind of relationship to the capacity c(u,v).

88. What is the min-cut corresponding to the max-flow in this graph?



89. Write the correct residual network for the flow network shown below?



90. Write the definitions for the following terms

    (a) dynamic threading

    (b) static thereading

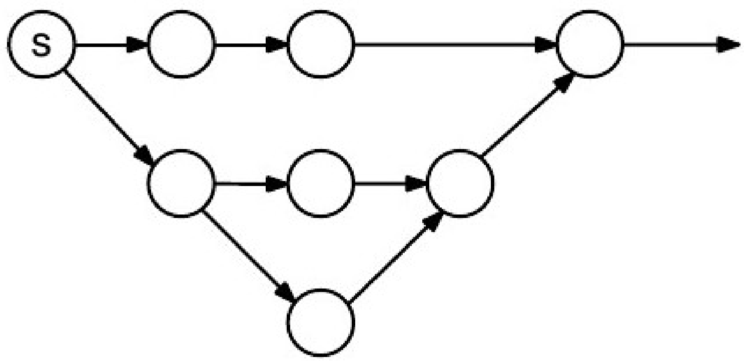
    (c) race condition

    (d) strand

    (e) dependency

91. Which keywords can a compiler **always** safely ignore?

92. Consider the following multithreaded computation DAG where the computation starts at s and each vertex represents a task that takes 1 unit of time to execute.



    Write the span and work of this computation.

93. The work of an algorithm is T_1 = n log n. The span is T_∞ = log n. What is the maximum number of processors for which linear speedup is possible?

94. Suppose we have shown that f(n) = O(g(n)) and g(n) = Θ(h(n)). Then which of these are necessarily true?

    - g(n) = O(f(n))
    - f(n) = O(h(n))
    - h(n) = $\Theta(g(n))$
    - g(n) = $\Omega(f(n))$
    - f(n) = o(f(n))

95. Suppose that G* = (V, E*) is the transitive closure of directed graph G. Suppose (v,w) is in E* and (u,v) is in E^*. What other edge must be in E^*?

96. If we implement Dijkstra's algorithm with the binary min-heap priority queue and then run it repeatedly to solve the all-pairs shortest paths problem on a connected graph G = (V,E), then (express runtimes in terms of both E and V, assuming G is connected but make no assumptions about sparseness):

    - Runtime for each pass on Djikstra's is
    - Required time to call Dijkstra's is
    - solution to the all-pairs shortest paths problem costs
    - This does not work on what kind of weights

97. In Johnson's algorithm, the addition of a new vertex s to G with outgoing edges to all other vertices adds new cycles to G and eliminates negative weight cycles.

    TRUE or FALSE

98. Which of these sets consist only of problems that are known to be verifiable (but not necessarily solvable) in polynomial time?

    (a) problems in P
    (b) problems in NP
    (c) problems in NP-Complete
    (d) problems in NP-Hard
    (e) Unsolvable problems

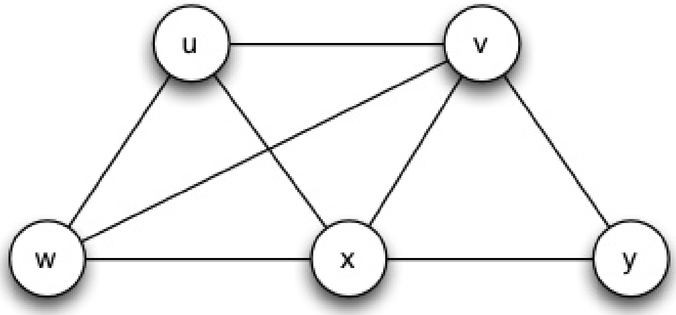
99. Which of these sets consist only of problems that are known to be solvable in polynomial time?

    (a) problems in P
    (b) problems in NP
    (c) problems in NP-Complete
    (d) problems in NP-Hard
    (e) Unsolvable problems

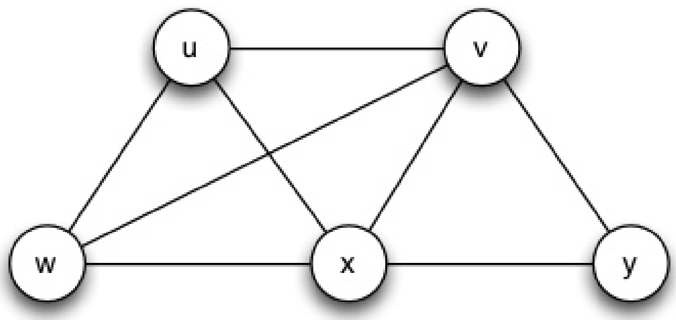
100. Suppose I wanted to show that the problem of optimal iThingy Configuration is NP Hard. I have to map:

101. We have seen that finding shortest paths in a graph can be solved in polynomial time, but the lecture notes say that the problem of finding longest paths is NP-Hard.
     If longest paths is NP-Hard, what is a plausible reason why were are able to find them in polynomial time for the job scheduling problem that we did in class?

102. Identify a clique of maximum size in the following graph



103. Which of these are a minimum vertex cover for the graph below?



104. Find the runtime of the following

```
function FOO(n)
    if n < 2
        return 1
    else
        for i = 1 to n
            x = x + 1
        for i = 1 to 2
            x = x + FOO(n/4)
        return x
```

105. $5n^8 + 2^n = \Theta(n^8)$

TRUE or FALSE

106. $n^9 = o(9^n)$

TRUE or FALSE

107. $n^5 \log n = \Omega(n^5)$

TRUE or FALSE

108. Given a recurrence relation that describes its worst case running time and give its worst case running time in Θ for the following
    (a) Binary Search
    (b) Insertion Sort
    (c) Merge Sort

109. Suppose that an array contains n numbers, each of which is -1, 0, or 1. Then, the array can be sorted in O(n) time in the worst case.

TRUE or FALSE

110. Heapsort can be used as the auxiliary sorting routine in radix sort, because it operates in place.

TRUE or FALSE

111. Suppose that a hash table of m slots contains a single element with key k and the rest of the slots are empty. Suppose further that we search r times in the table for various other keys not equal to k. Assuming simple uniform hashing, the probability is $\frac{r}{m}$ that one of the r searches probes the slot containing the single element stored in the table

TRUE or FALSE

112. Does there exist a polynomial algorithm that finds the value of an s-t minimum cut in a directed graph?

113. Given C is the cost of the approximate solution and C^* the cost of the optimal solution, which definition of the approximation ratio $\rho(n)$ would you use for a MINIMIZATION problem?

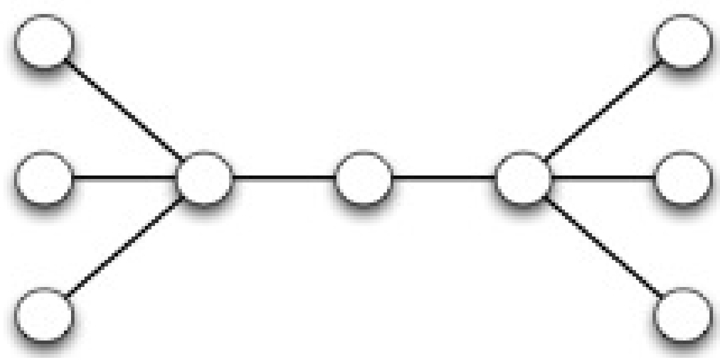
114. Match the approximation method to the NP-Hard problem for which a polynomial time 2-approximation algorithm was presented in the text and lecture.

    (a) MAX-3-CNF satisfiability
    (b) Traveling salesperson problem
    (c) Minimum weight vertex cover
    (d) Set Cover

115. Assuming that P $\neq$ NP, if I can use an approximation algorithm to find a solution to an NP-Hard problem that is within a factor of 2 of optimal, then can I use polynomial time reductions to find solutions to all NP-Hard problems within a factor of 2 of optimal?

116. The triangle inequality is how we know that when APPROX-TSP-TOUR is applied to a graph for which distances meet the triangle inequality,

117. Consider this graph and this algorithm:

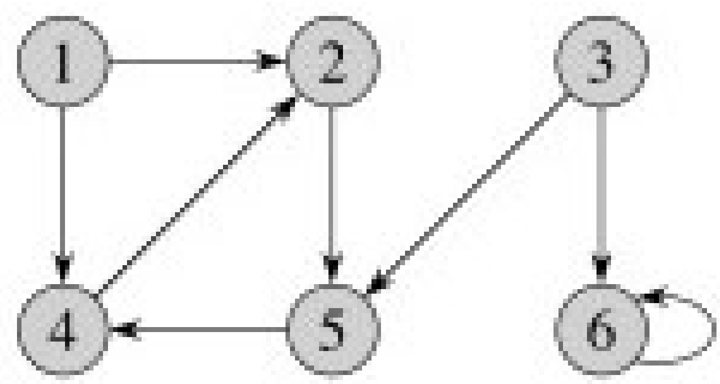


    (a) How many vertices are in the optimal vertex cover for this graph?
    (b) Assuming that line 4 always makes the best possible choice, what is the smallest number of vertices that APPROX-VERTEX-COVER can possibly select as a vertex cover for the above graph?
    (c) In general, APPROX-VERTEX-COVER is a _ approximation algorithm for Vertex Cover.

118. Write the Graph Search to the characterization of its process

    (a) Depth First Search
    (b) Breadth First Search

119. Show the d and π values that result from running BFS (as given in the CLRS text) on the following directed graph G, starting with s = vertex 3. Use "NIL" or "nil" as they do, and "INF" or "inf" for infinity. Be careful to respect the direction of arcs. Take your time to get it right!



120. What is the asymptotic run time of BFS in terms of V and E?

121. Given a "classic" adjacency list representation of a directed graph how long does it take to compute and print a table of the in-degree of all the vertices in the graph?

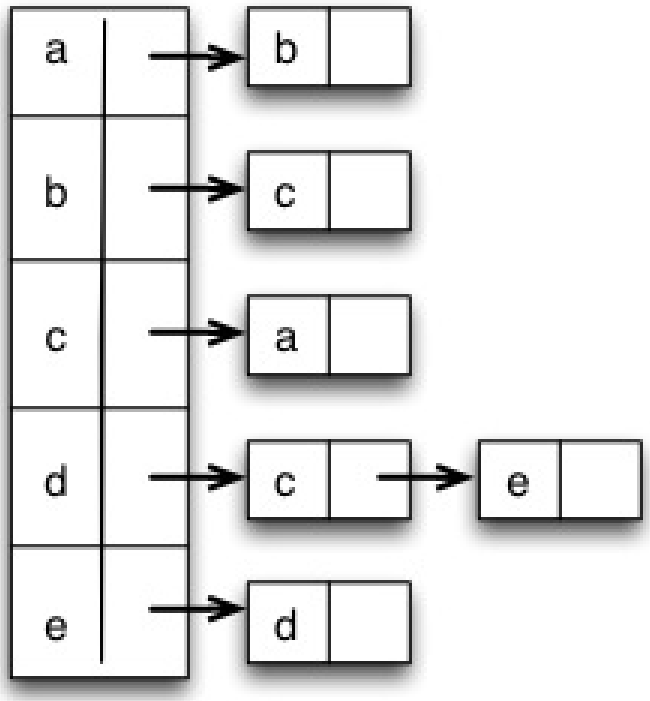
122. Given a "classic" matrix representation of a directed graph how long does it take to compute and print a table of the in-degree of all the vertices in the graph?

123. What are true of the accounting method of analysis?

124. The expected height of a skip list of n items

125. The expected number of items scanned at each level of a skip list of n items is O

126. Conduct a DFS (using the algorithm given by CLRS) on the graph shown in its adjacency list representation. For each vertex v show the discovery time v.d and the finish time v.f. Start with vertex a and process vertices in G.V and Adj[u] as if they are in alphabetical order.



127. Identify the back-edge(s) found by the search above.

128. Identify the cross-edge(s) found by the search above.