Growth of functions and Asymptotic analysis

1. Simplify the following expression: $\sum_{i=0}^{n} 2^i$.

2. In O notation, give the asymptotic bound for $\sum_{k=1}^{n} 1/k$.

3. Simplify the following expression $\sum_{k=0}^{n} k$.

4. Which of O, Ω or Θ is the correct asymptotic relationship for $f(n) = 3n + \sqrt{n}$ compared to $g(n) = n \ln n$?

5. Which of O, Ω or Θ is the correct asymptotic relationship for $f(n) = 1/n^{1.0} + 1/n^{-0.1}$ compared to $g(n) = 1/n$?

6. Which of O, Ω or Θ is the correct asymptotic relationship for $f(n) = 2^n$ compared to $g(n) = 2^{2n}$?

7. Which of O, Ω or Θ is the correct asymptotic relationship for $f(n) = n\sqrt{n}$ compared to $g(n) = n \ln n$?

8. Which of O, Ω or Θ is the correct asymptotic relationship for $f(n) = \sqrt{n}$ compared to $g(n) = \log^2(n)$?

9. Which of O, Ω or Θ is the correct asymptotic relationship for $f(n) = (n+2)(3n^2 - 3)$ compared to $g(n) = n^3$?

10. Which of O, Ω or Θ is the correct asymptotic relationship for $f(n) = 2^n$ compared to $g(n) = 2^{n+7}$?

11. Sudoku is a game played on a 9x9 board, in which each column, each row, and each of the nine 3x3 subboards must contain all numbers 1–9 exactly once. What is the asymptotic running time of an algorithm that enumerates all possible Sudoku boards that satisfy these constraints?

12. A train leaves Chicago at 4:20pm and travels south at 25 miles per hour. Another train leaves New Orleans at 1:23pm and travels north at 50 miles per hour. The conductors of both trains are playing a game of chess over the phone. After each player moves, the other player must move before their train has traveled 5 miles. In asymptotic notation, how many moves do the two players make before their trains pass each other (somewhere near Memphis)?

Recurrence relations

13. True or false: If the running time of an algorithm satisfies the recurrence relation $T(n) = T(n/10) + T(9n/10) + c\,n$, then $T(n) = O(n^2)$.

14. What is the asymptotic solution to the recurrence relation $T(n) = T(n/2) + c$, where c is a constant?

15. What is the asymptotic solution to the recurrence relation $T(n) = 2\,T(n/2) + c\,n$, where c is a constant?

16. What is the asymptotic solution to the recurrence relation $T(n) = T(n/3) + c$, where c is a constant?

17. Consider the following function:

```
def foo(n) {
     if (n > 1) {
        print( ''hello'' )
        foo(n/4)
        foo(3n/4)
}}
```

In asymptotic notation, give the number of times "hello" is printed, as a function of n.

18. Using the master method, give the asymptotic solution to the recurrence relation $T(n) = 9T(n/3) + n$?

19. Using the master method, give the asymptotic solution to the recurrence relation $T(n) = T(2n/3) + 1$?

20. Using the master method, give the asymptotic solution to the recurrence relation $T(n) = 3T(n/4) + n \log n$?

21. Using the master method, give the asymptotic solution to the recurrence relation $T(n) = 8T(n/2) + n^2$?

Binary search and QuickSort

22. What is the best-case asymptotic running time of binary search on a sorted array A containing n elements?

23. What is the worst-case asymptotic running time of QuickSort on an array A containing n elements?

24. True or false: the running time of the `Partition` function in QuickSort is linear in the size of the subproblem.

25. Suppose that 47.2% of the time the `Partition` function in QuickSort chooses a pivot element that is the median of the particular subproblem it is sorting, while the other 52.8% of the time, it chooses the minimum value. What would be the asymptotic running time of QuickSort in this case?

26. Using the input array $A = [9, 7, 5, 11, 12, 2, 14, 3, 10, 6]$ and the QuickSort algorithm, what is the number of times a comparison is made to the element with value 3. (Assume the partition algorithm always chooses the last element as the pivot.)

27. True or false: Divide and conquer algorithms only work when a problem has optimal substructure.

28. True or false: The standard way to prove the correctness of a divide and conquer algorithm is through a loop invariant.

Dictionaries

29. Suppose we implement a dictionary ADT using a standard linked list. What is the asymptotic running time of the `Find` operation?

30. What is the worst-case asymptotic running time for deleting an element from a hash table containing n items?

31. What is the expected running time to returning the minimum value stored in a hash table with ℓ locations and n values?

32. Write down an array with 5 elements that is a max heap, but whose reverse is not a min heap.

33. HeapSort is a sorting algorithm that inserts n items into a standard binary min-heap, and then pops them off one at a time, in ascending order. In asymptotic notation, how much space does HeapSort require to sort n elements?

34. In asymptotic notation, how long does it take to find the minimum value in a binary search tree containing n elements?

35. Let H be a hash table with 2018 slots with a hash function $h(x)$ with the uniform hashing property. Given an item x, what is the probability it hashes to the 17th location?

36. Let H be a hash table with 2018 slots with a hash function $h(x)$ with the uniform hashing property. Given two items x, y, what is the probability that they hash to the same location?

37. Using asymptotic notation, for what value of the load factor α does a hash table become inefficient relative to a red-black tree?

38. Suppose we initialize an empty dynamically resizing hash table H, and each time every location contains k expected items (for k constant) we do the size-doubling operation on H. In asymptotic notation, what is the cost of inserting n items into H?

Greedy algorithms and Huffman encoding

39. True or false: greedy algorithms require a problem to have optimal substructure.

40. True or false: InsertionSort is a greedy algorithm.

41. True or false: greedy algorithms, if they are correct, are always optimal.

42. True or false: greedy algorithms are a special case of dynamic programming.

43. For an input symbol set Σ of n symbols, exactly how many internal nodes in the encoding tree does Huffman's algorithm create?

44. Under Huffman's algorithm, what is the exact worst-case length of the longest codeword for $|\Sigma| = n$ input symbols?

45. True or false: the set of codewords $\{00, 010, 011, 10, 01\}$ is a valid output of Huffman's algorithm.

46. True or false: the set of codewords $\{111, 110, 0, 100, 101\}$ is a valid output of Huffman's algorithm.

47. True or false: the set of codewords $\{01, 11, 100, 101\}$ is a valid output of Huffman's algorithm.

48. For $\Sigma = \{a, b, c\}$, give the set of inequalities on the frequencies f_a, f_b, f_c that would yield corresponding codewords of $\{0, 10, 11\}$ under Huffman's algorithm.

49. Suppose a message is composed of the following letters and corresponding frequencies: $\{(A : 6), (L : 1), (G : 4), (O : 5), (S : 2)\}$. How many bits does Huffman's algorithm

use to encode the symbol G?

Dynamic programming

50. True or false: dynamic programming is a kind of greedy algorithm.

51. For two strings x, of length n_x, and y, of length n_y, give the asymptotic running time for computing the alignment cost matrix via dynamic programming.

52. Given two strings x, of length n_x, and y, of length n_y, and the alignment cost matrix S, what is the exact length of the longest possible sequence of edit operations that can be extracted from S.

53. Given two strings x, of length n_x, and y, of length n_y, and the conventional edit operations and costs, what is the maximum number of *swoperations* (transpositions) in any optimal alignment?

54. True or false: dynamic programming works best when solutions to subproblems can be memoized.

55. True or false: Memoization is an algorithmic strategy that reduces running time at the cost of additional space

56. True or false: Memoization always uses at most a polynomial amount of additional space.

57. Given two strings x, of length n_x, and y, of length n_y, and the alignment cost matrix S, what is asymptotic running time of extracting an optimal alignment?

58. Given two strings x, of length n_x, and y, of length n_y, and the alignment cost matrix S, what is asymptotic running time of counting the number of optimal alignments?

59. Using dynamic programming, state the asymptotic running time of calculating the nth Fibonacci number, defined as $F_n = F_{n-1} + F_{n-2}$.

Graphs, BFS, and DFS

60. Let A be the adjacency list of a simple graph $G = (V, E)$. What is the asymptotic running time of converting A into an adjacency matrix representation of G?

61. Let A be the adjacency list of a simple graph $G = (V, E)$, and let B be a $V \times V$ matrix of zeros. What is the asymptotic running time of converting A into an adjacency matrix representation in B?

62. State the asymptotic running time of Breadth First Search when applied to a simple graph $G = (V, E)$.

63. True or false: BFS enqueues vertices at the end of its queue and dequeues them from the head of its queue, while DFS enqueues at the end of its queue and also dequeues from the end.

64. True or false: The worst-case running time of BFS is better than that of DFS.

65. True or false: There exists a family of graphs $G = (V, E)$, in which V can be of arbitrary size and G is connected, on which DFS and BFS will visit vertices in the same order, for some choice of initial vertex $v \in V$.

66. True or false: The expected worst-case running time of "random graph search" (where elements are dequeued randomly) is better than the worst-case running time of BFS.

67. In exact terms of the degree of a vertex k, what is the maximum number of times BFS will dequeue a vertex i?

Single-source shortest path algorithms

68. When running Dijkstra's algorithm on a graph G with no negative-weight edges, what is the maximum number of times an edge may be relaxed?

69. When running Dijkstra's algorithm on a graph G with no negative-weight edges, what is the maximum number of edge relaxations that may occur before the algorithm halts?

70. Using a standard binary heap in Dijkstra's algorithm, what is asymptotic running time of generating a SSSP on a graph $G = (V, E)$ with no negative-weight edges?

71. Suppose $G = (V, E)$ has a weight function w that includes a least one negative-weight cycle. How many steps will Dijkstra's algorithm take before it halts?

72. Suppose $G = (V, E)$ has a weight function w that includes some negative-weight edges. After how many steps of running the Bellman-Ford algorithm would we know that G contains a negative-weight cycle?

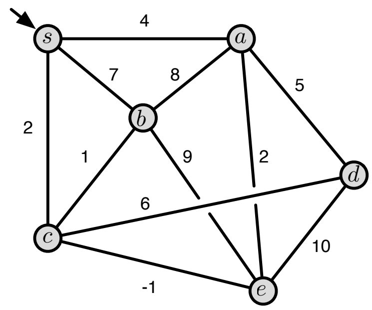
73. True or false: Given a graph $G = (V, E)$ with weight function w such that the most negative weight edge has weight w^*. Define a new weight function $w'(e) = w(e) + |w^*|$ on $e \in E$. If G contained no negative weight cycles under w, will Dijkstra's algorithm on G with w' produce the same output?

74. True or false: If G is an unweighted, undirected graph, every SSSP for G is also a MST for G.

75. Given the below undirected, weighted graph G, suppose we run a BFS starting from s. Write down the order in which the vertices are visited.

76. For the same graph as in question (75), suppose that we run Dijkstra's algorithm starting from s. Draw the SSSP tree that it produces.

77. For the same graph as in question (75), suppose that we run Dijkstra's algorithm starting from s. What is the 4th edge that Dijkstra relaxes to produce the SSSP tree.



This graph is used in questions (75), (76), (77), (84), (85), and (86).
Assume that G is stored as an adjacency list with adjacencies in lexicographic order.

Minimum spanning trees

78. Let G be a connected, undirected graph with 172 vertices and 291 edges. How many edges does a minimum spanning tree for G have?

79. Let $G = (V, E)$ and consider a minimum spanning tree T for G. What is the maximum number of safe edges that G may contain?

80. Let $G = (V, E)$, and consider a minimum spanning tree T for G. If G contains the minimum possible number of useless edges, how large is E?

81. What is the smallest number of phases in Boruvka's algorithm for obtaining an MST for a graph $G = (V, E)$?

82. What is the largest number of phases in Boruvka's algorithm for obtaining an MST for a graph $G = (V, E)$?

83. True or false: Kruskal's algorithm for obtaining an MST is a greedy algorithm.

84. For the same graph as in question (75), suppose that we run Kruskal's algorithm to obtain a minimum spanning tree for G. What is the 5th edge that Kruskal adds to the MST.

85. For the same graph as in question (75). Draw the MST for G.

86. True or false: The MST for the graph in question (75) is unique.

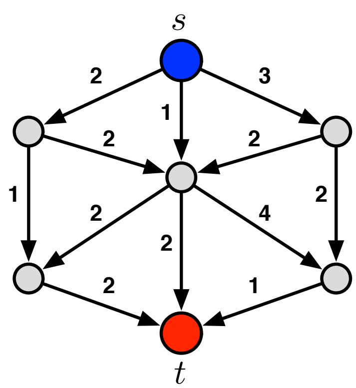
87. True or false: The MST for a graph is unique if and only if all the edge weights are distinct.

Max-flow and Min-cut

88. True or false: the Ford-Fulkerson algorithm for max-flow is a greedy algorithm, choosing the augmenting path with the largest additional flow F.

89. Let $G = (V, E)$ be a directed graph with positive integer edge capacities, let $s, t \in V$ be the source and sink vertices, let f^* be the value of the maximum flow on G, and

let c^* be the maximum edge capacity. If we are using the Ford-Fulkerson algorithm to solve an instance of max-flow, what is the worst-case number of augmenting paths we find before halting?



The graph used in questions (90), (91), (92), (93), and (94).

90. For the flow graph G above, what is the magnitude of the maximum flow f on G?

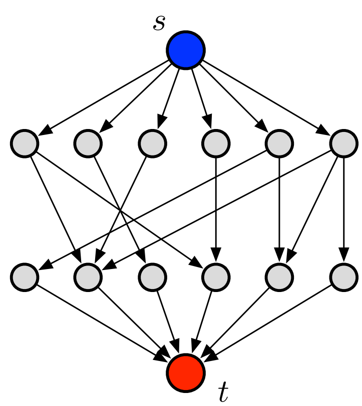
91. For the same graph as in question (90), find the cut that corresponds to the maximum flow. How many edges in this cut are saturated?

92. For the same graph as in question (90), what is the largest amount of additional flow that could be pushed from s to t if we are allowed to increase the capacity of any one edge by an arbitrary amount?

93. For the same graph as in question (90), what is the minimum number of augmenting paths necessary to achieve the maximum flow?

94. For the same graph as in question (90), at the initialization of a max-flow algorithm, what is the length (number of edges) of the longest augmenting path?

95. Given the following flow graph G, with unit capacities, consider the following task. A maximum flow on G is also a *maximum matching* on the bipartite subgraph G' composed of only the grey vertices. (Every grey vertex in the upper group can use at most one of its outgoing edges; and every grey vertex in the lower group can use at most one of its incoming edges.) What is the size of the maximum matching (flow) on G'?



The graph used in question (95); all edges have capacity of 1.

P and NP

96. True or false: every optimization problem has an equivalent decision problem.

97. True or false: P is a class of decision problems.

98. True or false: NP is a class of optimization problems.

99. True or false: The name of the complexity class NP is short for "Not Polynomial" time.

100. True, false, or an open question: P is a subset of NP.

101. True, false, or an open question: NP is a subset of P.

102. Given a simple graph $G = (V, E)$, what is the longest length of a Hamiltonian cycle?

103. Let T be a spanning tree for a graph $G = (V, E)$ with weight function w, and let k be the YES threshold for the decision version of the Minimum Spanning Tree problem. In asymptotic notation, how long does it take to verify that T is a witness for G?

104. Let f be a flow on a graph $G = (V, E)$ with edge capacity function c, and let k be the YES threshold for the decision version of the Max-Flow Problem. In asymptotic notation, how long does it take to verify that f is a witness for G?

105. How much money could you make if you could prove that P=NP?

106. True or false: A *reduction* shows that a problem of unknown difficulty is a special case of a known NP-complete problem.

107. True or false: Suppose that we are doing a reduction of problem $\mathcal{A}$ to problem $\mathcal{B}$, and it costs $O(n^c)$ for c constant to transform the input of $\mathcal{A}$ into an input of $\mathcal{B}$, and $O(2^{\ln n})$ to transform the output of $\mathcal{B}$ back into an output for $\mathcal{A}$. Is this reduction a polynomial-time reduction?

108. The problem Graph Isomorphism is a famous member of the complexity class NP but is not known to be a member of the NP-Complete set of problems within NP (and is believed not to be). Suppose someone announces that they have discovered a polynomial-time algorithm for solving Graph Isomorphism. Would this prove that P=NP?

109. Suppose someone announced that they'd discovered a way to solve Boolean Satisfiability (SAT) via a reduction from Hamiltonian Cycle (HC), and that the running time of converted an instance of SAT into HC took $O(1.00001^n)$ time, while the running time to convert the output of HC back into a solution for SAT took time $O(n^{1.00001})$. Would this prove that P=NP?

110. True or false: NP was defined to capture the notion of "problems solvable by brute force."

111. True or false: every problem solvable by a brute force algorithm is in NP.

112. True or false: An algorithm solving Hamiltonian Cycle with running time $O(n^{1,000,000})$ would imply P=NP.

113. True or false: An algorithm solving Hamiltonian Cycle with running time $O(n^{\log^* n})$ would imply P=NP.

114. True or false: every problem in NP is either in P or is NP-complete.

115. Give the name of one NP-complete problem.

116. True or false: all NP-complete problems are equivalent to one another.

117. True or False: Deciding if a graph has an Eulerian cycle is NP-complete.

118. True or False: Min-Cut is NP-complete if and only if P=NP.