

CS231 JEOPARDY: THE HOME VERSION*The game that turns CS231 into CS23fun!***Asymptotics**

 [1] Of the recurrence equations below, indicate

1. Which gives rise to the best asymptotic running time?
2. Which gives rise to the worst asymptotic running time?

- a. $T(n) = 1 + T(n - a)$, $a > 0$
 - b. $T(n) = 1 + T(an)$, $0 < a < 1$
 - c. $T(n) = n + T(n - a)$, $a > 0$
 - d. $T(n) = n + T(an)$, $0 < a < 1$
- -----

[2] In a graph $G = (V, E)$, under what conditions is $O(\sqrt{\lg(E)}) = O(\sqrt{\lg(V)})$?

[3] List all of the following sets that are subsets of $O(n^2)$

- | | | |
|-----------|-------------|-------------|
| a. $o(n)$ | f. $o(n^2)$ | k. $o(n^3)$ |
| b. $O(n)$ | g. $O(n^2)$ | l. $O(n^3)$ |
| c. (n) | h. (n^2) | m. (n^3) |
| d. (n) | i. (n^2) | n. (n^3) |
| e. (n) | j. (n^2) | o. (n^3) |
- -----

[4] List all of the following sets that are non-empty:

- a. $o(n)$ (n)
 - b. $o(n)$ (n)
 - c. $o(n)$ (n)
 - d. $o(n)$ (n)
 - e. $(n) - (o(n))$ (n)
- -----

[5] Consider the recurrence $T(n) = n + k(n/2)$. For each of the following sets, give a value of $k \geq 1$ such that $T(n)$ will have a solution in that set.

- a. (n)
- b. $(n \lg(n))$
- c. (n^2)

Sorting/Order Statistics

[1] What is the best worst-case running time for a comparison-based sort of n numbers?

[2] Consider sorting n numbers (not necessarily distinct) in the range $[1..n]$. List *all* of the following algorithms that have $O(n \lg(n))$ worst-case running times on this problem:

- a. selection sort
 - b. insertion sort
 - c. bubble sort
 - d. merge sort
 - e. quick sort (using two-finger partition with random pivot)
 - f. heap sort
 - g. tree sort (insert elements into binary search tree, and extract them via in-order traversal.)
 - h. counting sort
 - i. radix sort
 - j. general bucket sort
-
-

[3] The quicksort algorithms we studied in class had $(n \lg n)$ expected running time but (n^2) worst-case running time. Describe a single simple modification to these algorithms that makes them run in $(n \lg n)$ worst-case time

[4] Given a heap with n elements, what is the best-case running time for an algorithm that removes all the elements from the heap in sorted order?

[5] Using an $O(n)$ worst-case linear-time black-box subroutine for finding the median of a set of n numbers, describe a simple linear-time algorithm that solves the selection problem for an arbitrary order statistic.

Dynamic Sets

[1] For which of the following dynamic set implementations does the `Search` operation have an $O(\lg(n))$ worst-case running time?

- a. unsorted array
 - b. sorted array
 - c. unsorted singly-linked list
 - d. sorted singly-linked list
 - e. unsorted doubly-linked list
 - f. sorted doubly-linked list
 - g. binary tree
 - h. binary search tree
 - i. red-black tree
-

[2] List all of the following dynamic set implementations whose `Predecessor` operation has a smaller asymptotic worst-case running time than that for red-black trees :

- a. unsorted array
 - b. sorted array
 - c. unsorted singly-linked list
 - d. sorted singly-linked list
 - e. unsorted doubly-linked list
 - f. sorted doubly-linked list
 - g. binary tree
 - h. binary search tree
-

[3] What are the (1) worst-case and (2) expected running times of inserting n distinct elements in random order into the following dynamic set structures

	worst-case	expected
a. binary tree		
b. binary search tree		
c. red-black tree		

[4] For each of the following augmentations to each node of a red-black tree, indicate which *cannot* be maintained without changing the $O(\lg(n))$ worst-case running time of `Insert` and `Delete`:

- a. minimum of the subtree rooted at the node.
- b. successor of the node.
- c. size of the subtree rooted at the node.
- d. depth of the node.
- e. height of the node.
- f. black-height of the node.

[5] Give formulae for the (1) smallest and (2) largest possible number of internal (i.e., non-leaf) nodes in a red-black tree with black height k .

Graphs

[1] Suppose that G is a complete graph of cities whose edges indicate the distance between the cities. Match each of the following problems with the algorithms (there may be more than one) that can be used to solve it:

- a. Find the smallest length of fiber-optic cable to connect all the cities.
- b. Find the shortest distance path between two cities.
- c. Find the path between two cities that visits the fewest cities in-between.

1. Breadth-first Search
 2. Depth-first Search
 3. Prim's Algorithm
 4. Dijkstra's Algorithm
-
-

[2] List all of the following kinds of edges that cannot be found when performing depth-first search on an undirected graph:

- a. Forward edges
 - b. Back edges
 - c. Cross edges
-
-

[3] What is the name of a directed graph

- a. That has no back edges?
 - b. That has no back edges, forward edges, or cross edges?
-
-

[4] Draw the smallest graph for which depth-first search induces a different tree from breadth-first search.

[5] Say that a vertex A "spans" a vertex B if the discovery/finish time interval found by depth-first search for A includes the interval for B . Evaluate the following two claims:

- a. If a graph G contains a vertex V that spans all other vertices, then the graph has a single connected component.
 - b. If a graph G has a single connected component, then it must contain a vertex V that spans all other vertices.
-
-

Complexity

[1] For each of the following statements, indicate whether (1) it is known to be true (2) it is known to be false or (3) its truth is unknown.

- a. $P = NP$
- b. $P \neq NP$
- c. $P = \text{co-NP}$
- d. $P \neq \text{co-NP}$
- e. $NP = \text{co-NP}$
- f. $NP \neq \text{co-NP}$

[2] Suppose that A is a language in P . For each of the following conditions, indicate whether it is sufficient to prove that B is in P .

- a. $A \leq_P B$
- b. $B \leq_P A$
- c. $\overline{A} \leq_P B$
- d. $B \leq_P \overline{A}$

[3] Suppose that A is an NP-complete language and B is in NP. For each of the following conditions, indicate whether it is sufficient to prove that B is NP complete.

- a. $A \leq_P B$
- b. $B \leq_P A$
- c. For all L in NP, $L \leq_P B$

[4] Let L be the language $\{\langle A, k, n \rangle \mid n \text{ is the } k\text{th smallest element of array } A\}$. Using a black box algorithm A_L that decides L in polynomial time, show that there is a polynomial time algorithm that finds the k th order statistic of A in polynomial time.

[5] A simple path between vertices a and b of a graph G is said to be *hamiltonian* if it includes all the vertices in G . In the following, assume that G is a complete, weighted graph, and that a and b are arbitrary vertices in G . For each of the following languages, indicate whether it is "obviously" in each of P , NP, or co-NP:

- $L1 = \{\langle G, a, b, k \rangle \mid \text{there is a path between } a \text{ and } b \text{ whose weight is } < k.\}$
 - $L2 = \{\langle G, a, b, k \rangle \mid \text{there is a hamiltonian path between } a \text{ and } b \text{ whose weight is } < k.\}$
 - $L3 = \{\langle G, a, b, k \rangle \mid \text{all paths between } a \text{ and } b \text{ have weight } < k.\}$
 - $L4 = \{\langle G, a, b, k \rangle \mid \text{all hamiltonian paths between } a \text{ and } b \text{ have weight } < k.\}$
- -----

Potpourri

[1] Of the following greedy algorithms, list all that are *not* optimal:

- a. 0/1 knapsack algorithm
- b. fractional knapsack algorithm
- c. Prim's algorithm
- d. Kruskal's algorithm
- e. Dijkstra's algorithm
- f. coin-changing with arbitrary denominations
- g. Huffman coding

[2] Consider a game in which you flip two fair coins and are paid \$1 for each head and \$2 for each tail. What is the amount of money you expect to earn from playing this game once?

[3] True or false: A dynamic programming algorithm that uses a d-dimensional table, each of whose dimensions is size n, runs in $O(n^d)$ time. Briefly justify your answer.

[4] Calculate the sum $\sum_{k=1}^{20} (3 + 5k)$

[5] Using Huffman coding, what is the number of bits needed to encode the string ABRACADABRA. (Do *not* include the number of bits to encode the Huffman tree/table.)

