Algorithm Analysis Review

CS 111 – Fall 2015

Nov 19 Lecture

1. Sequential Search

What is the worst case running time of sequential search?

A. O(1)
B. O(n)
C. O(n * log n)
D. $O(n^2)$

2. Binary Search

Counting == as one comparison, and < as another comparison, how many comparisons would binary search make to find 65 in the array
[2 10 15 36 65 86 93] ?

a. 3
b. 4
c. 5
d. 6
e. None of the above

3. Selection Sort

The WORST case input for the selection sort algorithm is (if only comparisons count towards analysis):

a. Already sorted descending
b. Already sorted ascending
c. Random ordering
d. All cases require the same number of comparisons

4. Insertion Sort

The BEST case input for the insertion sort algorithm is (if only comparisons count towards analysis):

a. Already sorted descending
b. Already sorted ascending
c. Random ordering
d. All cases require the same number of comparisons

5. Insertion Sort

The WORST case input for the insertion sort algorithm is (if only comparisons count towards analysis):

a. Already sorted descending
b. Already sorted ascending
c. Random ordering
d. All cases require the same number of comparisons

6. Selection Sort sequence

Selection sort finds the max in every step and swaps with the last item in the current sub array. What is a possible step in sequence of sorting via selection sort for the following input: 4,2,3,10,5

a. 2,4,3,5,10
b. 4,2,3,5,10
c. 3,2,5,4,10
d. 5,3,2,5,10
e. None of the above

7. Insertion Sort sequence

What is a possible step in sequence of sorting via insertion sort for the following set: 4,2,3,10,5

a. 2,3,4,10,5
b. 5,2,3,4,10
c. 2,3,5,4,10
d. 10,3,2,5,4
e. None of the above

8. Running Time Orders

Match the alternative names for each of these running times:

$O(1), O(n), O(n^2), O(\log n), O(2^n)$

A. Quadratic, linear, constant, logarithmic, exponential
B. Linear, constant, exponential, logarithmic, quadratic
C. Constant, linear, quadratic, logarithmic, exponential
D. 1, enn, enn-squared, log enn, twototaen

9. Order from greatest to least running time

A. Exponential, quadratic, linear, logarithmic, constant

B. Linear, quadratic, constant, logarithmic, exponential

C. Quadratic, exponential, linear, constant, logarithmic

D. Constant, logarithmic, linear, quadratic, exponential

10. We want to calculate the class average of ONE assignment. Assume n=number of students, and g=number of assignments

The efficiency of the best algorithm to do this runs in this time:

A. O(g)
B. O(n)
C. O(n * g)
D. O(n/g)

11. We want to calculate the class average of ALL assignments. Assume n=number of students, and g=number of assignments

The efficiency of the best algorithm to do this runs in this time:

A. O(g)
B. O(n)
C. O(n x g)
D. O(n/g)

12. We want to find the common items in two unsorted lists, of lengths n and m, respectively.

The efficiency of the best algorithm to do this runs in this time:

A. O(n+m)
B. O(n * m)
C. O(n^2)
D. O(m^2)

13. We want to find the common items in two lists, of lengths n and m, respectively. The n list is unsorted, but the m list is sorted.

The efficiency of the best algorithm to do this runs in this time:

A. O(n+m)
B. O(n * m)
C. O(n * log m)
D. O((n+m)^2)

14. We want to fill all the cells of an integer n*m matrix with the value 10.

The running time to do this is:

A. Quadratic
B. Linear
C. Logarithmic
D. Exponential