

CS2223 D Term 2020 Quiz 5

(1 point) Question 1: “My brain is open...”

I pledge that I am taking this quiz on my own, with help from no one else and no notes:

(3 points) Question 2: INSERTIONSORT and SELECTIONSORT are both $O(n^2)$ in worst case because each makes $O(n)$ passes over the data to be sorted doing $O(n)$ comparisons on each pass, but INSERTIONSORT (unlike SELECTIONSORT) is $O(n)$ in the best case because:

- a.) In the best case, INSERTIONSORT does no comparisons on each pass.
- b.) In the best case, INSERTIONSORT does $O(1)$ comparisons on each pass.
- c.) In the best case, INSERTIONSORT does $O(n)$ comparisons on each pass.
- d.) In the worst case, SELECTIONSORT does no comparisons on each pass.
- e.) In the worst case, SELECTIONSORT does $O(n)$ swaps on each pass.

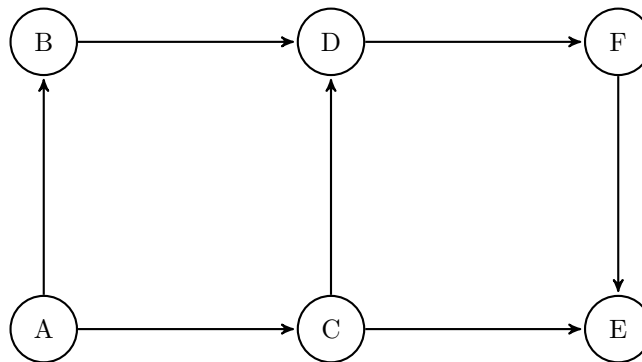
(3 points) Question 3: (1, 3, 6, 2, 4, 12, 8, 24)

The above linear ordering of the divisors of 24 does not respect the “divisor” relation and hence is not a legitimate topological sorting because:

- a.) 2 is a smaller prime divisor than 3 and so should be to the left of 3.
- b.) 2 is a divisor of 6 and so should be to the left of 6.
- c.) 8 has just four divisors; 12 has six divisors and so should be left of 8.
- d.) All the powers of 2 should be consecutive: (1,3,6,12,2,4,8,24) respects “divisor”.
- e.) Not all of the divisors of 24 are present.

(3 points) Question 4: Which of the following represents an order in which the vertices in the digraph below might be sorted topologically, if we think of the digraph as a representation of scheduling/precedence?

Hint: Break possible “ties” with lexicographical ordering.



- a.) A B C D E F
- b.) A B C D F E
- c.) A B D F E C
- d.) A C D E B F
- e.) A C E B D F

(1 point) Bonus Question: Both INSERTIONSORT and SELECTIONSORT are $O(n^2)$ in the average case—data to be sorted presented in random order. Which algorithm do you expect to have shorter running time in the average case?

- a.) SELECTIONSORT will be faster because it makes $O(1)$ swaps per pass instead of the $O(n)$ of INSERTIONSORT.
- b.) INSERTIONSORT will be faster because it makes no swaps on each pass in best case.
- c.) INSERTIONSORT will be faster because it will perform on average half as many comparisons per pass as SELECTIONSORT.
- d.) INSERTIONSORT will be faster because its best case is $O(n)$.
- e.) There will be no difference because both have the same $O(n^2)$ run time complexity.