# 1. CMPS 101 — Algorithms and Abstract Data Types

**References:**

1. Baase and Van Gelder: *Computer Algorithms*, 3rd Ed. Addison Wesley, 2000.
2. Cormen, Leiserson, Rivest: *Introduction to Algorithms, 2nd Ed.*. McGraw-Hill, 2001.

**Scoring:** Each question will have four choices for the answer, $(a)$–$(d)$. You will earn two (2) points for each correct answer. A blank answer counts zero; a wrong answer counts –1. You will have one–two "grace" questions that you can leave blank and still earn a perfect score.

The example problems here are indicative of the topics and format for the exam. The actual questions will be variations with slightly different problem instances, numbers, and mathematical expressions. However, the **concepts** that are tested by these questions will not change on the exam.

**Problems:**

1. [2] $6n^2 \log n + 2n + 5$ is in:
   (a) $\Theta(n^2)$.
   (b) $O(n^2)$.
   (c) $O(n^3)$.
   (d) $\Omega(n^3)$.

2. [2] $O(\sqrt{n})$ includes the functions:
   (a) $\frac{1}{2}\sqrt{n}$ and $0.001\,n$.
   (b) $2\sqrt{n}\ \log n$ and $2\sqrt{n}$.
   (c) $\frac{1}{2}\sqrt{n}\ \log n$ and $\frac{1}{2}\sqrt{n}$.
   (d) $2\sqrt{n}$ and $10\log n$.

3. [2] If $f(n)$ is in $\Theta(n^E)$, then:
   (a) $f(n)$ is in $O(n^E)$ but not in $\Omega(n^E)$.
   (b) $f(n)$ is in $\Omega(n^E)$ but not in $O(n^E)$.
   (c) $f(n)$ is in $O(n^E)$ and also in $\Omega(n^E)$.
   (d) All of the above are possible, depending on $f$ and $E$.

4. [2] If $f(n)$ is in $\Theta(g(n))$ and $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ exists, then *some possible* values for this limit are:
   (a) 10 and 100.
   (b) 0 and 100.
   (c) 10 and $\infty$.
   (d) 0, 1 and $\infty$.

5. [2] $\sum_{k=1}^{n} k \log_2(k)$ is in:
   (a) $\Theta(n \log(n))$.
   (b) $\Theta(k \log_2(k))$.
   (c) $\Theta(n^2 \log(n))$.
   (d) $\Theta(k^2 \log_2(k))$.

6. [2] $\sum_{i=0}^{n} i^2$ is in:
   (a) $\Theta(n^4)$.
   (b) $\Theta(n^3)$.
   (c) $\Theta(n^2 \log n)$.
   (d) $\Theta(n^2)$.

7. [2] $\sum_{j=1}^{K} 3^j$ is in
   (a) $\Theta(\frac{1}{4}K^4)$
   (b) $\Theta(\frac{1}{2}K^2)$
   (c) $\Theta(2 \cdot 3^K)$
   (d) $\Theta(3 \cdot 2^K)$

8. [2] $2n \log n + 2n + 5$ is in:
   (a) $\Omega(n^2)$.
   (b) $O(n^2)$.
   (c) $O(n)$.
   (d) $\Theta(n)$.

9. [2] If $f(n)$ is in $\Theta(n^E)$, then:
   (a) $f(n)$ is in $O(n^E)$ and also in $\Omega(n^E)$.
   (b) $f(n)$ is in $\Omega(n^E)$ but not in $O(n^E)$.
   (c) $f(n)$ is in $O(n^E)$ but not in $\Omega(n^E)$.
   (d) All of the above are possible, depending on $f$ and $E$.

10. [2] $O(\sqrt{n})$ includes the functions:
    (a) $2\sqrt{n}$ and $10 \log n$.
    (b) $2\sqrt{n} \log n$ and $2\sqrt{n}$.
    (c) $\frac{1}{2}\sqrt{n} \log n$ and $\frac{1}{2}\sqrt{n}$.
    (d) $\frac{1}{2}\sqrt{n}$ and $0.001 n$.

11. [2] $\sum_{i=2}^{2n} i$ is in:
    (a) $\Theta(n)$.
    (b) $\Theta(n^2)$.
    (c) $\Theta(n^2 \log n)$.
    (d) $\Theta(n^3)$.

12. [2] $\sum_{j=1}^{2\log_2(n)} 2^j$ is in
    (a) $\Theta(2^j)$
    (b) $\Theta(2^n)$
    (c) $\Theta(n^2)$
    (d) $\Theta((\log_2(n))^3)$

The next two questions are about the recurrence equation $T(n) = 2T(n/3) + n$.
Note that $log_2(3) \approx 1.6$ and $1/log_2(3) \approx 0.6$.

13. [2] To try to apply the Master theorem, one calculates that:
    (a) $E = 1$ and $f(n) = n^1$.
    (b) $E \approx 1.6$ and $f(n) = n^1$.
    (c) $E \approx 0.6$ and $f(n) = n^1$.
    (d) $E = 1$ and $f(n) \approx n^{1.6}$.

14. [2] How does the Master Theorem apply to this problem?
    (a) The Master Theorem does not apply to $T(n) = 2T(n/3) + n$.
    (b) The Master Theorem shows that $T(n) \in \Theta(n \log n)$.
    (c) The Master Theorem shows that $T(n) \in \Theta(n^{0.6})$ (approx).
    (d) The Master Theorem shows that $T(n) \in \Theta(n)$.

15. [2] A sorting algorithm whose average case and worst case are in different asymptotic classes (different big-$\Theta$ classes) is:
    (a) insertion sort.
    (b) quick sort.
    (c) merge sort.
    (d) heap sort.

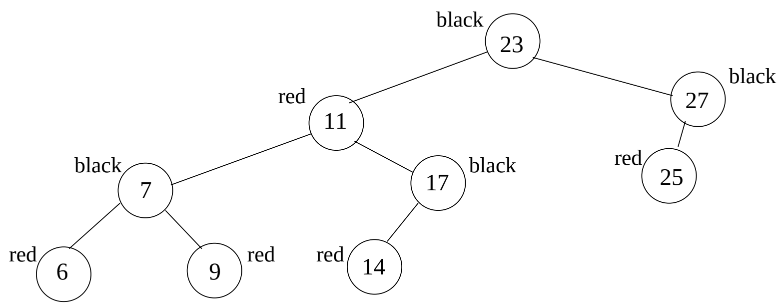
16. [2] A sorting algorithm that requires the most work space among the choices is:
    (a) insertion sort.
    (b) quick sort.
    (c) merge sort.
    (d) heap sort.

17. [2] A sorting algorithm whose best case is in the lowest asymptotic class (big-$\Theta$ class) among the choices is:
    (a) insertion sort.
    (b) quick sort.
    (c) merge sort.
    (d) heap sort.

18. [2] Two comparison-based sorting algorithms whose worst cases are asymptotically optimal are:
    (a) radix sort and heap sort.
    (b) quick sort and merge sort.
    (c) merge sort and heap sort.
    (d) quick sort and heap sort.

19. [2] The lower bound for all comparison-based sorting algorithms tells us that, in the worst cases:
    (a) Five elements cannot be sorted with six comparisons.
    (b) Five elements can be sorted with eight comparisons.
    (c) $n$ elements can be sorted with $n \log_2(n)$ comparisons.
    (d) Radix sort must use at least $n \log_2(n)$ comparisons, even if they are disguised as bit operations.

The following questions are based on the Red-Black tree shown below, using the standard Red-Black tree insertion algorithm and the standard binary search tree deletion algorithm.



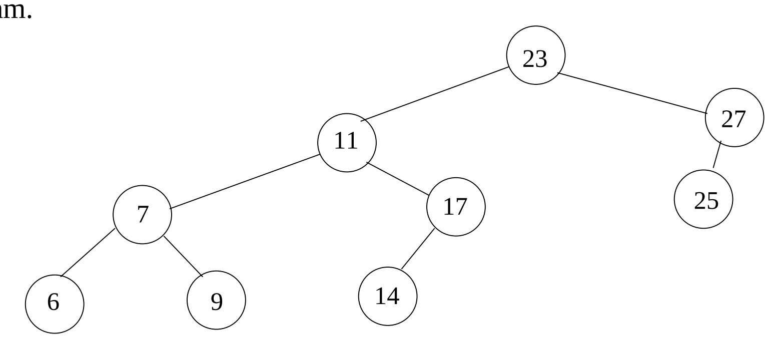

20. [2] After inserting 8 into the tree, what node is at the root?
    (a) 25
    (b) 23
    (c) 17
    (d) 11

21. [2] After inserting 8 into the tree, which nodes are black?
    (a) 11, 6, 9, 17, 27.
    (b) 7, 8, 11, 9, 14, 23.
    (c) 7, 8, 14, 23, 25.
    (d) 11, 6, 9, 7, 27, 25.

22. [2] What is the worst-case time complexity for an insert operation on a red-black tree with $n$ nodes?
    (a) $\Theta(1)$
    (b) $\Theta(\log n)$
    (c) $\Theta(n \log n)$
    (d) $\Theta(n^2)$

23. [2] What is the worst-case time complexity for a delete operation on a red-black tree with $n$ nodes?
    (a) $\Theta(1)$
    (b) $\Theta(n)$
    (c) $\Theta(\log n)$
    (d) $\Theta(n \log n)$

The following questions are based on the binary search tree tree shown below, using the standard binary search tree deletion algorithm.




24. [2] After deleting 23 with the standard binary-search-tree algorithm, what node is at the root?
   (a) 11
   (b) 25
   (c) 27
   (d) 14

25. [2] After deleting 23 with the standard binary-search-tree algorithm, what *parent* $\rightarrow$ *child* node pair **does not** occur in the tree?
   (a) 25 $\rightarrow$ 27
   (b) 27 $\rightarrow$ 11
   (c) 11 $\rightarrow$ 7
   (d) 7 $\rightarrow$ 9