

CSCI 33500: Software Analysis and Design III

Summer 2025

Midterm Review

1 TRUE OR FALSE

1. little o notation is used to represent an algorithms best case performance
2. when do you need to write the big 5 ?
3. The time complexity of search, insertion, and deletion operations in an AVL tree is $O(\log n)$.
4. An algorithm with time complexity $O(n!)$ is more efficient than one with time complexity $O(n \log n)$.
5. The Big Five in C++ include the default constructor, copy constructor, move constructor, copy assignment operator, and destructor.
6. The compiler distinguishes between the move constructor and the copy constructor based on whether the argument is an lvalue or an rvalue.
7. The space complexity of the selection sort algorithm is $O(1)$
8. If a function $f(n) \in \Theta(n \log n)$, then $f(n) \in O(n \log n)$ and $f(n) \in \Omega(n \log n)$.
9. In a proof by induction, the base case is optional as long as the inductive step is valid.
10. AVL tree insertions may trigger a rotation to maintain the balance condition.

2 SHORT ANSWER

1. $x = 49 \bmod 9$. assume x is greater than or equal to 0 and less than 9, what is x ?
2. solve $\gcd(318, 1002)$
3. Which proof technique would you use for the following question:
show that

$$\sum_{k=0}^n x^k = \frac{x^{n+1} - 1}{x - 1}$$

4. Write the code for an ll rotation in an avl tree.
5. is the following function declaration correct?

```
bool my_func(int val1=12,int val2);
```

6. Give the summation series for: 15, 21, 27, 33, ..., 177
7. Simplify the following function using big O notation. $3\frac{n}{7} + \log(n) + 3m + 1000000$
8. Given the following code, what will be printed ?

```
void PrintString(string& s)
{
    cout<<"l value function "<<s<<endl;
}

void PrintString(string&& s)
{
    cout<<"r value function "<<s<<endl;
}

int main()
{
    string s = "whatsup";
    PrintString("all");
    PrintString(s);
    PrintString(s+"all");
}
```

9. prove the following, if x is even then x^2 is also even.
10. what does `std::move` do ?

3 CODING

1. Write a recurrence relation for the power set (the power set is the set of all subsets for a given set), then compute the big O from your recurrence relation. Using vectors write the code to print the power set for any given set of size 3, what is the big O, big Ω and Big Θ for the code you wrote?
2. Given selection sort do an average time complexity analysis.

```
void selection_sort(int* arr, int size)
{
    for(int i=size-1;i>0;i--)
    {
        int max_index=0;
        for(int j=1;j<=i;j++)
            if(arr[j]>arr[max_index]) max_index=j;
        std::swap(arr[i],arr[max_index]);
    }
}
```