

CS 2009 – Design and Analysis of Algorithms

Fall 2024

September 5, 2024

Assignment #1 – All AI and DS Sections

Deadline: 9th September 2024 (8:30 AM to 9:30 AM)

Submission Venue: Office: A304a

Instructions:

1. **Guidelines Compliance:** Any violation of the provided guidelines will result in a score of zero for the assignment.
2. **Handwritten Submission:** The assignment must be completed in your handwriting.
3. **No Email or Soft copy Submissions:** Under no circumstances will submissions via email or digital format be accepted. All assignments must be submitted in handwritten format only.
4. **Individual Work:** This is an individual assignment. Collaboration or copying will not be tolerated.
5. **Assignment Distribution:** Students with even roll numbers should work on odd-numbered code segments, and those with odd roll numbers should work on ~~even~~-numbered code segments.
6. **No Late Submissions:** Late submissions will not be accepted.
7. **Identification Details:** Ensure that your Roll Number, Section, Name, and Signature are written at the bottom of each page. Failure to do so will result in mark deductions, and no subsequent queries about this matter will be entertained.
8. **Code Complexity Analysis:**
 - Perform a detailed complexity analysis for each provided code segment.
 - Include Time Complexity for each code.
 - Break down the complexity step-by-step, including the analysis of loops, and any nested structures.
 - If there are multiple solutions or approaches, discuss the complexities of each and provide a rationale for which approach is more efficient.
9. **Clarity and Neatness:** Your handwriting must be legible and neat. Any illegible or messy submissions may result in mark deductions.
10. **Reference Material:** While you may consult textbooks or other resources for your understanding, ensure all work is done in your own words.
11. **Additional Sheets:** You may attach extra sheets if required to complete your analysis or explanations. Ensure that any additional sheets also contain your Roll Number, Section, Name, and Signature at the bottom.

Note: Please adhere strictly to these instructions to ensure your assignment is graded fairly. Good luck and put forth your best effort in understanding and analyzing the complexities of the algorithms.

Code Segments for Complexity Analysis

1. Code Segment 1

```

1 for i = 1 to n;
2 {
3     for j = i to n
4     {
5         sum += a[i][j]
6     }
7 }
```

2. Code Segment 2

```

1 while (n > 0)
2 {
3     ans += n;
4     n /= 2;
5 }
```

3. Code Segment 3

```

1 for (i = n; i >= 1; i /= 2):
2 {
3     for j = m to i
4     {
5         ans += (i * j);
6     }
7 }
```

4. Code Segment 4

```

1 cout << "Hello world";
```

5. Code Segment 5

```

1 for i = 1 to n
2 {
3     for j = n to i
4     {
5         ans += (i * j);
6     }
7 }
```

6. Code Segment 6

```

1 for( i = 0; i < n; i++)
2 {
3     for( j = 0; j < 5; j += 2)
4         statements; // worth O(1)
5 }
```

7. Code Segment 7

```

1 for( i = n; i >= 0; i--)
2 {
3     for( j = 0; j < n; j++)
4         statements; // worth O(1)
5 }
```

8. Code Segment 8

```

1 // n > 0
2 k = 0;
3 while(1)
4 {
5     for(i = 0; i < n; i++)
6         continue;
7
8     if(k == !n)
9         break;
10 }
```

9. Code Segment 9

```

1 // n > 0
2 do
3 {
4     n--;
5 } while(n < 0);
```

10. Code Segment 10

```

1 int operator ++(int &n)
2 {
3     for(int i = 1; i <= x; i++)
4     {
5         cout << " ";
6     }
7     n++;
8     return n;
9 }
10 int main()
11 {
12     for(int i = 0; i < n; ++i)
13     {
14         continue;
15     }
16 }
```

11. Code Segment 11

```

1 sum = 0;
2 for( i = 1; i <= n * n; i++)
3     sum++;
```

12. Code Segment 12

```

1 sum = 0;
2 for( i = 1; i < n; i++)
3 ( j = 1; j < i * n; j++)
4 if( j % 1 == 0 )
5 ( k = 0; k < j; k++)
```

13. Code Segment 13

```

1 sum = 0;
2 for( i = 1; i < n; i++)
3     for( j = 0; j < i * n; j++)
4         ( k = 0; k < j; k++)
5             sum++;
```

14. Code Segment 14

```
sum = 0;
for ( i = 1; i <= n; i++ )
    for ( j = 0; j < i; j++ )
        sum++;
```

15. Code Segment 15

```
1 sum = 0;
2 for ( i = 1; i <= n * n; i++ )
3     for ( j = 0; j < n; j = j + 1 )
4         sum++;
```

16. Code Segment 16

```
1 sum = 0;
2 for ( i = 1; i < n; i = i * 2 )
3     for ( j = 0; j < n; j++ )
4         sum++;
```

17. Code Segment 17

```
1 void fun(int n)
2 {
3     int i, j, k, count = 0;
4     for ( i = n / 2; i <= n; i++ )
5         for ( j = 1; j + n / 2 <= n; j++ )
6             for ( k = 1; k <= n; k = k * 2 )
7                 count++;
8 }
```

18. Code Segment 18

```
1 Sum()
2 {
3     for i = 1 to n do
4         i = i + 1;
5     for j = 1 to n do
6         for k = n down to 1 do
7             sum = sum + 1;
8 }
```

19. Code Segment 19

```
1 int a = 0;
2 for ( i = 0; i < N; i++ ) {
3     for ( j = N; j > i; j-- ) {
4         a = a + i + j;
5     }
6 }
```

20. Code Segment 20

```
1 for ( i = n ; i > 0; i-- )
2 {
3     for ( j = 0; j < n; j++ )
4     {
5         cout << j;
6     }
7 }
```

21. Code Segment 21

```
1 for ( i = n ; i > 0; i-- )
2 {
3     for ( j = 0; j < n; j * 2 )
4     {
5         cout << i;
6     }
7 }
```

22. Code Segment 22

```
1 int total = 0;
2 for ( int i = 1; i <= n * n; i++ )
3 {
4     total++;
5 }
```

23. Code Segment 23

```
1 int m = 0;
2 for ( int i = 1; i <= n; i = 2 )
3 {
4     for ( int j = 0; j < n; j++ )
5     {
6         m++;
7     }
8 }
```

24. Code Segment 24

```
1 int a = 0;
2 for ( i = 0; i < n; i++ )
3 {
4     for ( j = n; j > i; j-- )
5     {
6         a = a + i + j;
7     }
8 }
```

25. Code Segment 25

```
1 while ( low <= high )
2 {
3     mid = ( low + high ) / 2;
4     if ( target < list [ mid ] )
5         high = mid - 1;
6     else if ( target > list [ mid ] )
7         low = mid + 1;
8     else break;
9 }
```


26. Code Segment 26

```

1 int key, j;
2 for (int i = 1; i < size; i++)
3 {
4     key = array[i];
5     j = i;
6     while (j > 0 && array[j-1] > key)
7     {
8         array[j] = array[j-1];
9         j--;
10    }
11    array[j] = key;
12 }

```

27. Code Segment 27

```

1 int sum = 0;
2 for (int i = 1; i <= n * n; ++i)
3 {
4     for (int j = 0; j < n; ++j)
5     {
6         ++sum;
7     }
8 }

```

28. Code Segment 28

```

1 int i, j, imin;
2 for (i = 0; i < size-1; i++)
3 {
4     imin = i;
5     for (j = i + 1; j < size; j++)
6     {
7         if (array[j] < array[imin])
8             imin = j;
9     }
10    swap(array[i], array[imin]);
11 }

```

29. Code Segment 29

```

1 for (i = 0; i < n; i++)
2 {
3     i *= 3;
4 }

```

30. Code Segment 30

```

1 long double sum = 0;
2 for (int i = 1; i <= n; ++i)
3 {
4     for (int j = 1; j < i * n; ++j)
5     {
6         if (j % 1 == 0)
7         {
8             for (int k = 0; k < j; ++k)
9             {
10                ++sum;
11            }
12        }
13    }
14 }

```

31. Code Segment 31

```

1 int sum = 0;
2 for (int i = 0; i <= n; ++i)
3 {
4     for (int j = 0; j < n; ++j)
5     {
6         ++sum;
7     }
8 }

```

32. Code Segment 32

```

1 for (i = 0; i < n; i++)
2 {
3     m += j;
4     m += j;
5     m += j;
6
7     m += j; // 31 times
8 }

```

33. Code Segment 33

```

1 for (i = 0; i < n; i++) {
2     subtotal = 0;
3     for (j = 0; j < i; j++)
4         subtotal += j;
5     tot += subtotal;
6 }

```

34. Code Segment 34

```

1 for (i = 0; i < n; i++)
2     for (j = 0; j < sqrt(995); j++)
3         m += j;

```

35. Code Segment 35

```

1 for (i = 0; i < n; i++)
2     for (j = 0; j < sqrt(n); j++)
3         m += j;

```