# CS203 B Data Structure

# Quiz 1

## Part 1 choices (50%)

1. Given a 2D array A of the integer ADT. It starts from A(-1, 3). The address of A(0,5) is 36, the address of A(1, 4) is 56, and the address of A(2, 7) is 92. What is the address of A(4, 4)?
   (a) 120
   (b) 128
   (c) 68
   (d) 76
2. The Ackermann function A(m, n) is defined recursively for non-negative integers m and n as follows:

$$\mathrm{A(m,n)} = \begin{cases} n+1, if \;\; m=0 \\ A(m-1,1) if \;\; m>0 \;\; and \;\; n=0 \\ A\big(m-1, A(m,n-1)\big), if \;\; m>0 \;\; and \;\; n>0 \end{cases}$$

   Its value grows very quickly, even for small values of m and n. For instance, A(4,1) = 65533. What would be the value of A(2,2)?
   (a) 4
   (b) 5
   (c) 6
   (d) 7
3. Consider the following code segment. (Assume that x and n are both positive and won't cause overflow.)

```
long foo (long x, long n) {
        long f;
        n % 2 == 0 ? f = 1 : f = x;
        if (n < 2 ) return f;
        return f * foo (x*x, n/2)
}
```

   Which of the following statement is true?
   (a) foo(x,n) return $x^n$
   (b) foo(x,n) return $n^x$
   (c) This recursive method runs in θ(log x) time
   (d) This recursive method runs in θ(nlog n) time

4. Please order the following function by growth rate in increasing order asymptotically. N, SQRT(N), $N^{1.5}$, $N^2$ , NlogN, Nlog(logN), $N\log^2 N$, $N\log(N^2)$, $2^n$, 29, $N^2\log N$, $N^3$. What is the function in the 5th place.
   (a) $N\log(N^2)$
   (b) NlogN
   (c) $N^{1.5}$
   (d) SQRT(N)

5. $T(n) = 4T\left(\frac{n}{2}\right) + n, T(1) = 1$. T = θ(?)
   (a) θ(n log n)
   (b) θ(n)
   (c) $\theta(n^2)$
   (d) $\theta(n^2 \log n)$

6. Most programming languages allocate storage for arrays sequentially and require all array elements to be of the same type, to allow efficient calculation of storage addresses of elements at run time. Suppose the Pascal array below is allocated sequentially in storage, in row major order, with 4 bytes per element, and starting address b.
   var A: array[1…5, 1 …10] of Real;
   The expression to calculate the storage address, in bytes, of element A[I,J] is:
   (a) $b + 4 \times (I - 1) + 20 \times (J - 1)$
   (b) $b + 40 \times (I - 1) + 20 \times (J - 1)$
   (c) $b + 40 \times I + 4 \times J - 44$
   (d) $b + 20 \times I + 4 \times J - 24$

7. Which of the following operation is not O(1) for any sorted data. You may assume that array elements are distinct.
   (a) Find the ith lagest element
   (b) Delete an element
   (c) Find the ith smallest element
   (d) All of the above

8. Let A be a square matrix of size n x n. Consider the following program. What is the expected output?

```
C = 100
for i =1 to n do
        for j =1 to n do
        {
                Temp = A[i][j] + C
                A[i][j] = A[j][i]
                A[j][i] = Temp - C
        }
for i =1 to n do
        for j =1 to n do
                output(A[i][j])
```

(a) The matrix A itself
(b) Transpose of matrix A
(c) Adding 100 to the upper diagonal elements and subtracting 100 from diagonal elements of A
(d) None of the above

9. Let A[1…n] be an array of n distinct number. If i < j and A[i] > A[j], then the pair(i,j) is called an inversion of A. What is the expected number of inversions in any permutation on n elements?
(a) n(n-1)/2
(b) n(n-1)/4
(c) n(n+1)/4
(d) n(n+1)/2

10. A program P reads in 500 integers in the range [0..100] exepresenting the scores of 500 students. It then prints the frequency of each score above 50. What would be the best way for P to store the frequencies?
(a) An array of 50 numbers
(b) An array of 100 numbers
(c) An array of 500 numbers
(d) A dynamically allocated array of 550 numbers

## Part 2 Essay(50%)

1. (12%) Consider the following algorithm.

```
compute(n,t)
{
    if (n <= 1) return atom(n-t)
    else{
        compute X, Y, and Z based on n
        s = 0
        for i = 1 to X
            do s = s + compute(Y, x*t)
        for z = 1 to Z
            do s = s + atom(x+t)
        return s
    }
}
```

Assume that the function atom as well as all the arithmetic operations in the algorithm takes $\theta(1)$ running time. In addition, assume that computing the values of X, Y and Z takes $\theta(1)$ running time. Determine the tightest asymptotic upper bound for the running time of compute(n,1) for different cases of (X,Y,Z).

| X | Y | Z | Running time of compute(n,1) |
|---|---|---|---|
| 5 | $\left\lfloor \frac{n}{4} \right\rfloor$ | $\lfloor n\sqrt{n} \rfloor$ | (a)G |
| 4 | $\left\lfloor \frac{n}{2} \right\rfloor$ | $n^2$ | (b)H |
| 3 | $\left\lfloor \frac{n}{3} \right\rfloor$ | $\lfloor nlogn \rfloor$ | (c)L |
| 2 | $n-2$ | $n$ | (d)E |

Please choose from the following items as the answers:

| | | | |
|---|---|---|---|
| (A) $O(1)$ | (B) $O(n)$ | (C) $O(\log n)$ | (D) $O(n\log n)$ |
| (E) $O(2^n)$ | (F) $O(n^2)$ | (G) $O(n\sqrt{n})$ | (H) $O(n^2\log n)$ |
| (I) $O(n \times 2^n)$ | (J) $O(\sqrt{n})$ | (K) $O(n^{\log_4 5})$ | (L) $O(n\log^2 n)$ |

2. For the following program fragment, give an analysis of the running time. (in big-O notation) (7%)

```
sum = 0;
for (i = 1; i<n;i++)
      for (j = 1; j< i * i ; j++) {
            sum++;
            for ( k = 0 ; k < j ; k++)
                  sum++;
      }
```

$O(n^5)$

3. Solving the recurrence $T = 2T(\lfloor\sqrt{n}\rfloor) + \log n, T(1) = 1$, using big-O notation as tight as possible. (7%)

   T(n) = O(log(logn)*logn)

4. Define ADT. (5%)

   An abstract data type(ADT) is a data type that is organized in such a way that the specification of the objects and the operations on the objects is separated from the representation of the objects and the implementation of the operations.

5. X is a 2-dimension array. The address of X(4,2) is 1978 and X(2,3) is 1986. Assume a byte machine is used. Each element of X occupies two bytes.

   (a) X(3,8) = ? (4%)

   2048

   (b) #rows of X = ? (4%)

   6

   (c) row major or column major? (1%)

   column major

6. Naïve matrix multiplication of two $n$ x $n$ matrices, A and B, runs in $\theta(n^3)$ time. Strassen's algorithm is a divide-and-comquer approach for matrix multiplication, By partitioning A and B into four $n/2$ x $n/2$ matrices respectively, instead of 8 multiplications, Strassen showed a method that only requires 7 matrix multiplications and additional $\theta(n^2)$ scalar operations to compute A x B.

   (a) Show that Strassen's algorithm runs is $\theta(n^{log_2 7})$ (5%)

   (b) Dr. Q proposed a method, called Algorithm Q, that can multiply two $c$ x $c$ matrices with $m_c$ multiplications and $\theta(n^2)$ scalar operations for a constant $c$. (For Strassen's algorithm, $c = 2$ and $m_2 = 7$.) Determine the asymptotic complexity of Algorithm Q for two $n$ x $n$ matrices in terms of $n$, $c$ and $m_c$. You may assume that $m_c > c^2$ (5%)

(a) $T(n) = 7T(n/2) + \theta(n^2)$

By master method,

$a = 7, b = 2, f(n) = n^2$

$n^{log_2 7} > n^2$

$T(n) = \theta(n^{log_2 7})$

(b) $T(n) = m_c T(n/c) + \theta(n^2)$

因為 $m_c > c^2$, $n^{log_c m_c} > n^2$

$T(n) = \theta(n^{log_c m_c})$

## Part 3 Bonus (20%)

1. The following procedure recursively generates all the permutations of list[i] to list[n].

```
void perm (char *list, int i, int n)
{
        int j, temp;
        if (i = = n)
        {
                for (j =0; j <= n; j++)
                        printf("%c", list[j]);
                printf("        ");
        }
        else {
                for (j = i ; j <= n ; j++)
                {
                        ________(B1)________
                        ________(B2)________
                        SWAP(list[i], list[j], temp);
}}}
```

   (a) Blank (B1) in the algorithm above should be (5%)?

      SWAP(list[i], list[j], temp)

   (b) Blank (B2) in the algorithm above should be (5%)?

      perm(list, i+1, n)

2. 對本門課的建議 (10%)