

**1. (6 points) True or False**

The following questions are True or False questions, you should judge whether each statement is true or false.

*Note: You should write down your answers in the box below.*

Problem 2.1	Problem 2.2	Problem 2.3
T	F	T

- (a) (2') Queue is the common data structure for implementation of Breadth First Traversal.
- (b) (2') The degree and the depth of the root node are both zero in all tree.
- (c) (2') If  $a$  is an ancestor of  $b$ , then there is exactly one unique path from  $a$  to  $b$  in the tree.

**2. (10 points) Recurrence and the Master Theorem**

Given the recurrence  $T(n) = aT(n/b) + f(n)$  with  $T(1) = 1$ .

If the recurrence indicates a divide and conquer algorithm,

- (a) (4') The original problem of size  $n$  is divided into A subproblems and each subproblem has size D
  - A.  $a$
  - B.  $b$
  - C.  $n/a$
  - D.  $n/b$
  - E.  $f(n)$
- (b) (2')  $f(n)$  is the time complexity of B
  - A. Divide and Conquer
  - B. Divide and Combine
  - C. Conquer and Combine
- (c) (2') If  $(a, b, f(n)) = (2, 3, 3\sqrt{n})$ , then the solution to this recurrence is  $T(n) = \underline{\text{O}(n^{\log_3 2})}$ .
- (d) (2') If  $(a, b, f(n)) = \underline{(2, 2, n)}$ , then the recurrence indicates the **Merge Sort** algorithm covered in our lecture. The solution to this recurrence is  $T(n) = \underline{\text{O}(n \log n)}$ .

*Note: Write your answer for time complexity in asymptotic order form i.e.  $T(n) = O(g(n))$ .*

**3. (8 points) Counting Ternary Ascending Subsequence**

Liu Big God has found ternary ascending subsequence very interesting. Given an array  $\langle a_1, \dots, a_n \rangle$ , he wants to calculate the number of 3-tuples  $(i, j, k)$  such that  $1 \leq i < j < k \leq n$  and  $a_i < a_j < a_k$ . He has almost finished the implementation, with some blanks left for you.

```

struct element {
    int value; // value of this element.
    int left;  // number of elements whose index and value are both greater than this
                element's.
    int right; // number of elements whose index and value are both less than this
                element's.
};

void elements_merge(element *a, int l, int mid, int r) {
    ...
}

int merge_and_calculate(element *a, int l, int r) {
    if(l == r) return 0;
    int mid = (l + r) / 2, rtn = 0;
    rtn += merge_and_calculate(a, l, mid);
    rtn += merge_and_calculate(a, mid + 1, r);
    int p1 = l, p2 = mid + 1, p3 = l;
    // for 3-tuple (i, j, k), we enumerate every j and consider whether j is in the left
    // -subarray or in the right-subarray to calculate the crossing-mid 3-tuples
    // when j is in the left-subarray
    // all elements whose index is not less than b are greater than a[j].value
    for(int j = l, b = mid + 1; j <= mid; j++) {
        while(b <= r && a[b].value <= a[j].value) b++;
        rtn += _____;
        a[j].left += (_____);
    }
    // when j is in the right-subarray
    // all elements whose index is less than b are less than a[j].value
    for(int j = mid + 1, b = l; j <= r; j++) {
        while(b <= mid && a[b].value < a[j].value) b++;
        rtn += _____;
        a[j].right += (_____);
    }
    // call this function to merge two sorted subarrays a[l..mid] and a[mid+1..r] into
    // one sorted array a[l..r]. Elements are compared by 'value'.
    elements_merge(a, l, mid, r);
    return rtn;
}

```

}

Each of the following questions has exactly one correct answer. Write your answers in the table below.

(a) (2') Which of the following code block should be filled in the first blank?

- A. `a[j].left * (b - mid)`
- B. `a[j].right * (b - mid)`
- C. `a[j].left * (r - b)`
- D. `a[j].right * (r - b + 1)`

(b) (2') Which of the following code block should be filled in the second blank?

- A. `r - b + 1`
- B. `r - b`
- C. `b - mid`
- D. `b - mid - 1`

(c) (2') Which of the following code block should be filled in the third blank?

- A. `a[j].left * (b - 1)`
- B. `a[j].right * (b - 1)`
- C. `a[j].left * (mid - b + 1)`
- D. `a[j].right * (mid - b)`

(d) (2') Which of the following code block should be filled in the fourth blank?

- A. `mid - b + 1`
- B. `mid - b`
- C. `b - 1`
- D. `b - 1 + 1`

4. (2 points) Guess the average score ( $\in [0, 26]$ ) of this quiz.

4. \_\_\_\_\_