Midterm Exam of Data Structures (CSE, NTOU)

Student ID: _____

09:20–12:00, 28 October 2026; Room INS105

Name: _____

Note: Cell phones and any calculator are forbidden.

1. (20%) Convert the following infix expressions into (i) postfix forms and (ii) prefix forms.

   (a). (10%) `(a * b + c / (d - e)) - f * (g + h)`

   (b). (10%) `(a * b) - (c / d)`

2. (10%) Given the following recursive program (pseudo-code) of the Fibonacci number, please convert it into a non-recursive code using the **stack** data structure. The only loop in your program is "while stack is not empty".

```
int Fibo(int n) {
    if (n == 0) return 0;
    else if (n == 1) return 1;
    else return Fibo(n-1) + Fibo(n-2);
}
```

3. (10%) Ackerman's function $A(m,n)$ is defined as:

$$A(m,n) = \begin{cases} n+1, & \text{if } m = 0, \\ A(m-1,1), & \text{if } n = 0, \\ A\big(m-1, A(m,n-1)\big), & \text{otherwise.} \end{cases}$$

   This function grows very quickly for small values of m and n. Please compute the values $A(1,1)$ and $A(2,2)$.

4. (10%) **Counting node visits.** There are n nodes arranged into k disjoint, nonempty circular singly linked lists, where $1 \le k \le n$. The array `nodes[0..n-1]` contains a pointer to each of these nodes exactly once, in arbitrary order. Consider the following code. Assume that `count` does not overflow.

```
unsigned long long count = 0;

for (int i = 0; i < n; ++i) {
    Node *p = nodes[i];

    do {
        ++count;
        p = p->link;
    } while (p != nodes[i]);
}
```

```
typedef struct Node {
    int data;
    struct Node *link;
} Node;
```

   (a) (5%) Let n_j denote the number of nodes in the jth circular list, for $1 \le j \le k$. Express the final value of `count` in terms of $n_1, n_2, \ldots, n_k$. Justify your answer.

   (b) (5%) Use the Cauchy–Schwarz inequality to prove that `count` $\ge \dfrac{n^2}{k}$.

5. (10%) Please complete the following function of concatenating two singly linked lists pointed by `p1` and `p2`.

```
pNode concatenate (pNode p1, pNode p2) {
    if (!p1) return p2;
    if (!p2) return p1;
    /* complete the rest of the function */
}
```

```
typedef struct listNode *pNode;
struct listNode {
    char data[4];
    pNode link;
};
```

6. (10%) Let `A`$[0..j, 0..k]$ be a two-dimensional array. Each element of `A` requires **two** bytes. Suppose `A`$[2,2]$ is stored at 878, and `A`$[4,5]$ is stored at 920. Assuming that memory is addressed in bytes, and the array is stored sequentially in memory. This array is **row-major** order.

   (a). (5%) What is the value of k?

   (b). (5%) What is the address of `A`$[5,7]$?

7. (10%) Given the following matrix `int A[3][4]` with integer entries. We use the sparse matrix ADT to store `A`.
   (a). (5%) How many bytes are required to store `A` using the sparse matrix ADT? (*Hint:* Consider the following structure on the right side for the sparse matrix ADT.)
   (b). (5%) Please write the detail of A^{T} stored in the sparse matrix ADT.

$$A = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 8 \end{bmatrix}$$

```
typedef struct {
    int col;
    int row;
    int value;
} term;
term A[MAX_TERMS];
// array of terms
```

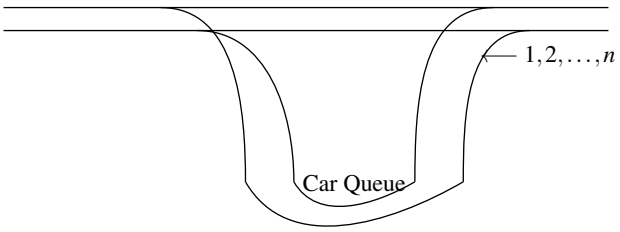
8. (8%) Given a string $s =$ "`ntountnu`", please compute the failure function values of each position (i.e., 8 integer values).

9. (12%) Order the following function by growth rate in increasing order (**Note:** $\log n =: \log_2 n = \lg n$) without proofs:

   (a). $(\lfloor \log n \rfloor)!$ (b). $\log(n!)$ (c). $2^{\sqrt{n}}$ (d). $4^{\log n}$

   Note: Your answer should be like "(a) < (b) < (c) < (d)". There are totally 6 pairwise relations; each accounts for (2%).

10. (5%) Consider the railroad switching network shown below. Cars numbered $1, 2, \ldots, n$ arrive from the right in that order, with car 1 arriving first. The lower track is an initially empty FIFO queue of **unlimited capacity**. Cars may also *bypass* the queue and move directly to the left output track. At each step, you may perform one of the following operations: "(1) Move the next input car to the rear of the queue; (2) Move the next input car directly to the output track, bypassing the queue; (3) If the queue is nonempty, remove its front car and move it to the output track." Cars cannot return to the input track or be moved back from the output track. The output permutation is the order in which cars enter the output track. For example, when $n = 2$, enqueue car 1, send car 2 directly to the output, and then dequeue car 1, producing $(2, 1)$. For $n = 3$, list all IMPOSSIBLE output permutations and briefly justify your answer.



11. (5%) Suppose that we present the polynomial ADT using singly the linked list. Please provide an instance to show that the adding two polynomials of m and n terms, respectively, exactly requires $m+n$ pointer movements. (**Note:** Count each execution of `a = a->link` or `b = b->link`, including the final advance to `NULL`.)

12. (10%) **True (T) or False (F).** Indicate whether each statement is true or false. Each correct answer accounts for **2%**.
   (a) For $n \geq 2$, if two $n \times n$ matrices each contain exactly n nonzero entries, their product contains at most n nonzero entries.
   (b) For compatible nonzero sparse matrices A and B, let R be the number of rows of A, C the number of columns of B, and t_A, t_B their respective numbers of nonzero entries. The sparse matrix multiplication has time complexity $O(Ct_A + Rt_B)$.
   (c) A circular array queue with N slots that reserves one slot to distinguish the full and empty states can hold at most $N-1$ elements.
   (d) Dequeuing an element from the circular array queue requires shifting every remaining element by one array position.
   (e) When searching for the same pattern in a different text, KMP must recompute the pattern's failure function.