※ 考生請注意：本試題不可使用計算機。 請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

Part I.

資料結構 (50%)

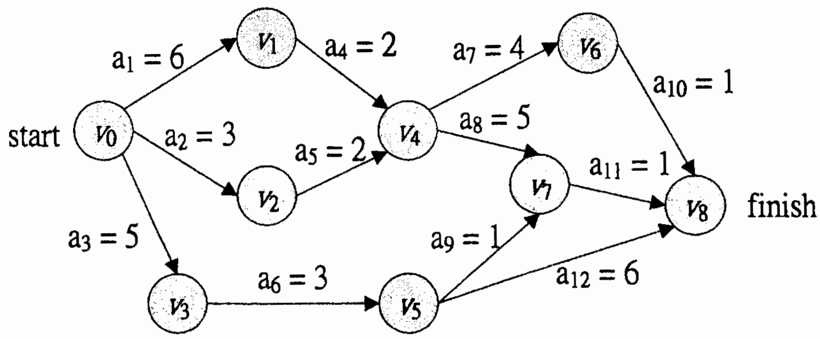
1. Answer **True** or **False** for the following statements. Give correct answers for **False** statements. (20%)
   (1) Let G be a graph with e edges and v vertices. If G is represented by adjacency lists, DFS requires $O(v^2)$ time.
   (2) If an AOV network represents a feasible project, its topological order is unique.
   (3) In static hashing, the worst-case number of comparisons needed for a successful search is $O(n)$ for open addressing. The number could be reduced to $O(\log n)$ by using chaining method.
   (4) Let d_i be the degree of vertex i in a graph G with $|V| = n$ and $|E| = e$, then $e = \sum_{i=0}^{n-1} d_i$.
   (5) The path from vertex u to vertex v on a minimal cost spanning tree of an undirected graph G is also a shortest path from u to v.

2. Given the following 8 runs: (15%)

7	10	23	2	6	4	16	12
13	18	28	15	11	20	17	21
22	29	32	19	24	30	31	25

   (1) Draw the corresponding winner tree.
   (2) Draw the restructured winner tree after one record has been output.
   (3) Draw the loser tree based on the answer of question (2).
   (4) Derive the total required time to merge n records through a winner tree with k runs.

3. Consider the following AOE network: (15%)



   (1) Obtain $e(i)$ and $l(i)$ for all activity i.
   (2) List all critical activities.
   (3) List all critical paths.

(背後仍有題目，請繼續作答)

※ 考生請注意：本試題不可使用計算機。　請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

Part II.

演算法 (50%)

1. Answer **True** or **False** for the following statements. Give correct answers for **False** statements. (20%)
   (1) If P equals NP, then NP equals NP-complete.
   (2) Any *n*-node unbalanced tree can be balanced using $O(\log n)$ rotations.
   (3) Let A_1, A_2, and A_3 be three sorted arrays of *n* real numbers (all distinct). In the comparison model, constructing a balanced binary search tree of the set $A_1 \cup A_2 \cup A_3$ requires $\Omega(n \log n)$ time.
   (4) Let P be a shortest path from some vertex s to some other vertex t in a graph. If the weight of each edge in the graph is increased by one, P remains a shortest path from s to t.
   (5) We can claim that there is a simpler way to reweight edges than the method used in Johnson's algorithm. Letting $w^* = min_{(u,v)\in E}\{w(u,v)\}$, just define $\widehat{w}(u,v) = w(u,v) - w^*$ for all edges $(u,v) \in E$.

2. Give a tight asymptotic upper bound (**O** notation) on the solution to the following recurrence. (10%)

$$T(n) = \begin{cases} 16T\left(\frac{n}{2}\right) + \Theta(1) & if\ n^2 > M, \\ M & if\ n^2 \le M; \end{cases}$$

3. Prove that the longest increasing subsequence problem can be reduced to the edit distance problem. (10%)
   Edit distance problem:
   Input: Two sequences A and B and the cost for operations substitution (C_s), insertion (C_i), and deletion (C_d)
   Output: The minimum cost sequence of edit operations to transform A into B.
   Longest increasing subsequence problem:
   Input: A sequence S.
   Output: The longest sequence of positions $\{p_1, p_2, \ldots, p_k\}$ such that $p_i < p_{i+1}$ and $S_i < S_{i+1}$.

4. Consider the linear-programming system with 9 different constraints $x_1 - x_5 \le -5, x_1 - x_4 \le -2, x_2 - x_1 \le -3, x_2 - x_3 = 8, x_3 - x_1 \le 5, x_3 - x_5 \le 2, x_4 - x_3 \le -3, x_5 - x_1 \le 6, x_5 - x_4 \le 1$, (1) Draw the constraint graph for these constraints. (2) Solve for the unknowns $x_1, x_2, x_3, x_4, and\ x_5$ or explain if no solution exists. (10%)