選擇題每一題組須全答對才計分。

Part I:Multiple Choice questions（選擇題(單選/多選)請使用答案卡作答）

1. (5%). Which of the followings are true? ①

(A) $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.
(B) $f(n) = o(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) \neq \Omega(g(n))$.
(C) $T(n) = 2 \cdot T(\sqrt{n}) + \Theta(\log n)$ has a solution $T(n) = O((\log n)^2)$.
(D) $T(n) = 8 \cdot T(n/3 + 12) + n^2$ has a solution $T(n) = \Theta(n^2)$.
(E) $T(n) = 2 \cdot T(n/4) + \sqrt{n}$ has a solution $T(n) = \Omega(\sqrt{n})$.

2. (4%). An $m \times n$ array A of real numbers is a *Monge array* if for any rows $1 \leq i < k \leq m$ and any columns $1 \leq j < \ell \leq n$, we always have

$$A[i,j] + A[k,\ell] \leq A[i,\ell] + A[k,j]. \qquad (1)$$

In other words, for the four entries at rows i, k and columns j, ℓ, the sum of the upper-left and lower-right elements is always at most the sum of the lower-left and upper-right elements. Consider the following two arrays

$$M_1 = \begin{bmatrix} 10 & 17 & 13 & 28 & 23 \\ 17 & 22 & 16 & 29 & 23 \\ 24 & 28 & 22 & 34 & 24 \\ 11 & 13 & 6 & 17 & 7 \\ 45 & 44 & 32 & 37 & 23 \\ 36 & 33 & 19 & 21 & 6 \\ 75 & 66 & 51 & 53 & 34 \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} 37 & 23 & 22 & 32 \\ 21 & 6 & 7 & 10 \\ 53 & 34 & 30 & 31 \\ 32 & 13 & 9 & 6 \\ 43 & 21 & 15 & 8 \end{bmatrix}$$

It can be verified that M_1 is a Monge array and M_2 is not. Which of the followings are true? ②

(A) To verify that the array A is a Monge array, it suffices to verify the inequality (1) for every rows $i, i+2$ and every columns j, ℓ, where $1 \leq i \leq m-2$ and $1 \leq j < \ell \leq n$.
(B) Changing the value of the element 7 in M_2 to 13 makes it a Monge array.
(C) To make M_2 a Monge array, at least two elements must change their values.
(D) Suppose that A is a Monge array and let $f(i)$ denote the column index of the leftmost smallest element at row i. Then $f(k) \geq f(i)$ for any $1 \leq i < k \leq m$.
(E) A is a Monge array if and only if for any $1 \leq i < m$ and $1 \leq j < l \leq n$, we always have
$$A[i,j] + A[i+1,\ell] \leq A[i,\ell] + A[i+1,j].$$

3. (6%) This problem considers permutations of $A_n = \{0,1,..,n-1\}$. Let P be a permutation from A_n. P can be represented as a 0-1 string S of length n-1, where:

- $S[i] ==$ '1' if $P[i] < P[i+1]$, and
- $S[i] ==$ '0' if $P[i] > P[i+1]$.

For example, a permutation $P = (0, 1, 3, 2)$ can be represented as a string ``110``. Note that this representation is not one to one, i.e., different permutations may be represented as the same string. Given a 0-1 string S, we are interested in reconstructing a possible permutation P.
Consider the following C++ procedure.

```
vector<int> findPermutation(string S) {
    int left = count(S.begin(), S.end(), '0'), right = left;
    vector<int> res = {left};
    for (char &c : S)
        res.push_back(c == '1' ? ++right : --left);
    return res;
}
```

Which of the following statements is/are correct? ③

(A) The call of findPermutation(``000``) returns the vector {3, 2, 1, 0}
(B) Permutation (3, 1, 2, 0, 4) can be represented as ``0101``, but it is not the corresponding output of the call of findPermutation(``0101``).
(C) findPermutation takes $O(n \log n)$ time to recover the permutation, where n is the length of the permutation.
(D) Not all the input can have an answer returned from the procedure findPermutation.
(E) The call of findPermutation(``01001``) returns the vector {3, 2, 4, 1, 0, 5}.

4. (4%) Let p be a prime number. Consider a positive integer a. We want to find another positive integer b such that $1 = ab \bmod p$. Let $p = 101, a = 55$. Which of the following statements about b is/are true? ④

(A) There are infinitely many $b's$ such that $1 = ab \bmod p$.
(B) $(b \bmod p)$ has 2 digits.
(C) $a < b$.
(D) $((b \bmod p) \bmod 10) > 3$.
(E) The digit sum of $(b \bmod p) > 10$.

5. (5%) Let A and B be two arrays and each has n positive numbers. Given a permutation π of {1, 2, ..., n}, we can compute the sum $\sum_{i=1}^{n} A[i] * B[\pi(i)]$. We are interested in finding the maximum of the sum, which can be maximized with a proper permutation π. Which of the following statements is/are true? ⑤

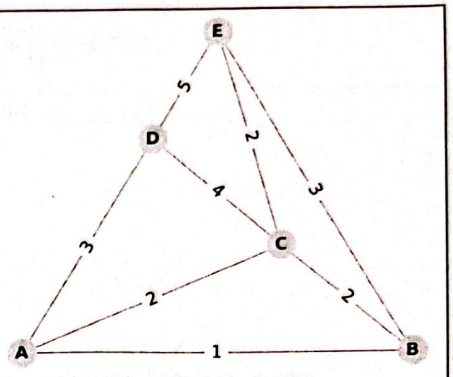
(A) If $a \leq b$ and $c \leq d$, then $a * c + b * d \geq a * d + b * c$.
(B) If A=[1, 3, 2, 4], B=[3, 4, 2, 1], then the maximum sum is 30.
(C) This problem can be solved in $o(n)$ time.
(D) This problem can be solved with the Greedy method.
(E) When the maximum is achieved, the corresponding permutation π will always make array B sorted.

6. (3%) Which of the following statements for minimum spanning tree (MST) are true? ⑥

(A) If an MST has N nodes, it has $N-1$ edges.
(B) If all edge weights in the graph are distinct, the MST is unique.
(C) If a constant C is added to the weights of all edges, the MST will remain the same.
(D) If the weight of a non-MST edge is decreased, the MST will not change.
(E) If the weight of a non-MST edge is increased, the MST will not change.

7. (3%) Given an undirected weighted graph $G = (V, E)$, where $V = \{A, B, C, D, E, F\}$, the edges and their corresponding weights are illustrated in the right figure. How many distinct minimum spanning trees (MSTs) can be formed from the graph G using Kruskal's or Prim's algorithm? ⑦

(A) 1
(B) 2
(C) 3
(D) 4
(E) 5



8. (3%) Which of the following problems can be solved using breadth-first search? ⑧

(A) Find the shortest path from the start to the end in a maze.
(B) Determine whether an undirected graph is a bipartite graph.
(C) In a directed network, starting from a specific node, find all nodes that can be reached from this node.
(D) In a tree or graph, find all nodes that are at a depth of k from a given starting node.
(E) Identify all connected components in an undirected graph.

9. (3%) Which of the following statements about Red-Black Trees are correct? ⑨

(A) A Red-Black Tree is a self-balancing Binary Search Tree.
(B) The length of any path from the root to a leaf differs by at most a factor of 2.
(C) The root node is red.
(D) If the deleted node is red, rebalancing is required.
(E) Left rotation and right rotation are used to restore the balance of the Red-Black Tree.

10. (3%) Considering the following implementation of min heap.

```
#define MAX_SIZE 10
int data[MAX_SIZE];
int size=0;
void insert(int x)
{
  if (size>=MAX_SIZE-1)   return;
  int cur= ____□□____ ;
  while(cur!=0 && data[____◇◇____]>x)   {
    data[cur]=data[____◇◇____];
    cur= ____◇◇____;
  }
  data[cur]=x;
}
```

Which way of filling in the code to make function insert work correctly? ⑩

(A) □□: size++, ◇◇: cur/2
(B) □□: ++size, ◇◇: cur/2
(C) □□: size++, ◇◇: (cur-1)/2
(D) □□: ++size, ◇◇: (cur-1)/2
(E) None of the above

11. (3%) If the following numbers are sequentially inserted into a min heap using the function insert,

40 15 50 10 30 20 5 12

what are the numbers of data[1] and data[4] (suppose that the index starts from 0)? ⑪

(A) data[1]: 12, data[4]: 30
(B) data[1]: 5, data[4]: 15
(C) data[1]: 10, data[4]: 20
(D) data[1]: 12, data[4]: 20
(E) None of the above

12. (3%) After sequentially inserting the following numbers into an empty B-tree with max degree 3 (i.e, each node has at most 3 children),

80 40 20 100 60 30 50 70 10 25 35

how many nodes in the B-tree are of one number? ⑫

(A) 4
(B) 6
(C) 8
(D) 10
(E) None of the above

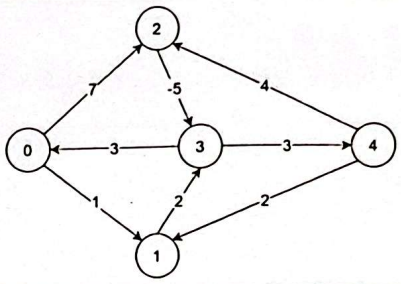
13. (3%) Which of the following statement(s) is/are true? ⑬

(A) AVL tree, B-tree and B+-tree are search trees.
(B) B+-tree offers more efficient sequential data access (such as "searching all numbers that are greater than 10") than B-tree does.
(C) B-tree offers more efficient data search than B+-tree does.
(D) AVL tree outperforms B-tree and B+-tree in minimizing the number of disk access operations.
(E) Consider the case that the following numbers are sequentially inserted into an empty AVL tree.

100 80 60 40 50 65 70 75 64

In the resultant AVL tree, the right child of root is 70.

14. (3%) Consider the right graph and the following functions. V is the number vertices, graph[V][V] is the adjacency matrix, and INF is the maximal number that can be stored in int. graph[u][v] is set to INF when there is no edge from u to v. Please answer the following question.____⑭____



(A) Dijkstra's algorithm can be used to get the shortest path from vertex 4 to all other nodes.
(B) Function foo1 can be used to get the shortest path from vertex 4 to all other nodes.
(C) Function foo2 can be used to get the shortest path from vertex 4 to all other nodes.
(D) Function foo3 can be used to get the shortest path from vertex 4 to all other nodes.
(E) Function foo2 is part of implementation of Dijkstra's algorithm.

```
void foo1(int graph[V][V], int src)
{
    int dist[V];
    for (int i = 0; i < V; ++i)
        dist[i] = graph[src][i];
    for (int k = 2; k < V; ++k)
        for (int u = 0; u < V ; ++u) {
            if (u == src) continue;
            for(int i=0; i < V; ++i)
              if (i!=u && graph[i][u] != INF &&
                    dist[i]!=INF &&
                    dist[u]>dist[i]+graph[i][u])
                dist[u]=dist[i]+graph[i][u];
        }
}

int minDistance(int dist[V], bool sptSet[V]);
void foo2(int graph[V][V], int src)
{
    int dist[V];
    bool sptSet[V];
    for (int i = 0; i < V; i++)
        dist[i] = INF, sptSet[i] = false;
    dist[src] = 0;
    for (int count = 0; count < V - 1; count++) {
        int u = minDistance(dist, sptSet);
        sptSet[u] = true;
        for (int v = 0; v < V; v++)
            if (sptSet[v]==false && graph[u][v]!=INF
                && dist[u] != INF
                && dist[u] + graph[u][v] < dist[v])
                dist[v] = dist[u] + graph[u][v];
    }
}
```

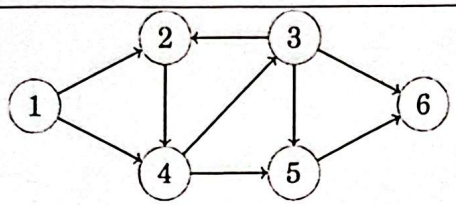
```
int minDistance(int dist[V], bool sptSet[V])
{
    int min = INF, min_index;
    for (int v = 0; v < V; v++)
        if (sptSet[v] == false && dist[v] <= min)
            min = dist[v], min_index = v;
    return min_index;
}
void foo3(int graph[V][V], int src)
{
    int i, j, k;
    int dist[V][V];
    for(i=0; i<V;i++)
        for(j=0;j<V;j++)
            dist[i][j]=graph[i][j];
    for (k = 0; k < V; k++) {
        for (i = 0; i < V; i++) {
            for (j = 0; j < V; j++) {
                if (dist[i][k] + dist[k][j] < dist[i][j])
                    dist[i][j] = dist[i][k] + dist[k][j];
            }
        }
    }
}
```

Part II: Non Multiple Choice questions（非選擇題）

1. (10%). You are given a directed acyclic graph $G = (V, E)$ with real-valued edge weights and two distinguished vertices $s, t \in V$. The weight of a path is the sum of the weights of the edges in the path.
   (a) Describe a dynamic-programming approach for finding a simple path from s to t with the maximum weight. Write down the recurrence formula you use and the definition of the subproblem(s) explicitly.
   (b) What is the running time of your algorithm?

2. (6%). A palindrome is a non-empty string over some alphabet that reads the same forward and backward. Examples of palindromes are all strings of length 1, *civic*, *racecar*, and *aibohphobia*. Give an efficient (polynomial-time) algorithm to find the longest palindrome that is a subsequence of a given input string. For example, given the input *character*, your algorithm should return *carac*. What is the time complexity of your algorithm?

3. (10%)
(a) (4%) Given a graph $G(V, E)$, show how to find the maximum number of **edge disjoint** paths from the source s to the sink t. This means that you should construct a set of paths such that each edge appears in at most one path. Consider the following example with $s = 1, t = 6$. What is the maximum number of edge-disjoint paths?



(b) (6%) Show how to find the maximum number of **node-disjoint** paths from the source to the sink. This means that you should construct a set of paths such that every node, except for the source and sink, may appear in at most 1 path.

4. (6%) A Red-Black Tree is a self-balancing binary search tree that maintains specific balance properties. For a given black height h, derive the minimum and maximum number of nodes $N_{\min}(h)$ and $N_{\max}(h)$ that a Red-Black Tree can have. Clearly explain the reasoning and provide the final formulas for $N_{\min}(h)$ and $N_{\max}(h)$ in terms of h.

5. (3%) In a heap of size n, what is the index range of the leaf nodes? Note that the index starts from 0.

6. (4%) A binary search tree can be augmented for specialized applications such as an order statistic tree. The provided pseudo-code is an implementation of the OS-SELECT function, which retrieves the i-th smallest element in the set represented by the binary search tree. The variable x represents a node in the tree and initially points to the root when the function is called. As the function recurses, x updates to reflect the current node being processed. Your task is to fill in the missing part of the pseudo-code in line 1.

```
OS-SELECT(x, i)
1   r ← _____
2       if i = r
3           then return x
4       else if i < r
5           then return OS-SELECT(left[x], i)
6       else return OS-SELECT(right[x], i − r)
```

7. (10%) Consider the following function implemented in C. Please analysis the time complexity of bar(N, 'A', 'B', 'C') in Big O notation using substitution method for solving recurrence relation. Please clearly show the steps of your analysis.

```
void bar(int N, char X, char Y, char Z)
{
  if (N<=1)
     printf("%c %c\n", X, Y);
  else   {
     foo(N-1, X, Z, Y);
     printf("%c %c\n", X, Y);
     foo(N-1, Z, Y, X);
     }
}
```