You must answer the questions *in order* on the answer sheet.

1. (15 points) Give asymptotically upper (big O) bounds for $T(n)$ in each of the following recurrences, where we assume that $T(n)$ is constant for small n. Make your bounds as tight as possible, and justify your answers.
   (a) (5 points) $T(n) = 2T(\sqrt{n}) + \lg n$
   (b) (5 points) $T(n) = T(n/2 + \sqrt{n}) + n$
   (c) (5 points) $T(n) = T(n-1) + 1/n$

2. (20 points) The binomial tree B_k is a rooted tree defined recursively. The binomial tree B_0 consists of a single node. The binomial tree B_k consists of two binomial trees B_{k-1} that are linked together: the root of one is the leftmost child of the root of the other. Answer the following two questions.
   Each solution can use at most *200 Chinese characters* or *200 English words*. *Over-sized description will be ignored.*
   (a) (10 points) For the binomial tree B_k, how many nodes are there at depth i for $i = 0, 1, \ldots, k$? Justify your answer.
   (b) (10 points) What is the maximum degree of any node in an n-node binomial tree? Justify your answer.

3. (15 points) We consider the following problems of undirected graphs.

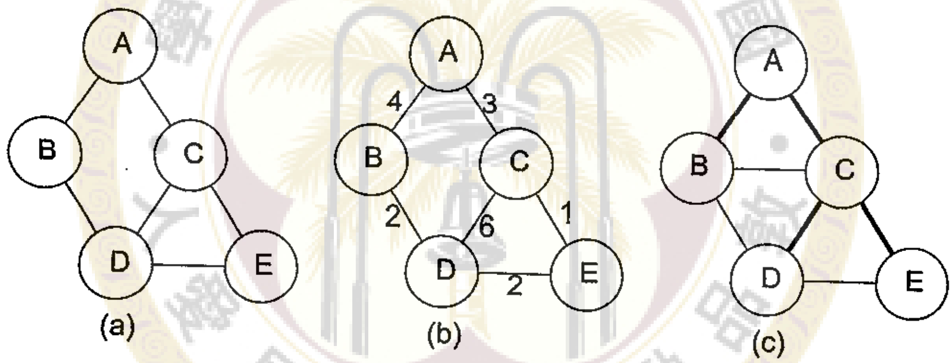

Figure 1: Three undirected graphs

Hamiltonian cycle problem is to determine whether a graph has a *Hamiltonian cycle*. A Hamiltonian cycle of a graph is a cycle that goes through every node of the graph exactly once. For example, A-B-D-E-C-A is a Hamiltonian cycle in Figure 1 (a).

Hamiltonian path problem is to determine whether a graph has a *Hamiltonian path*. A Hamiltonian path from a node a to a node b is a path from a to b that goes through all the other nodes of the graph exactly once. For example, E-D-C-A-B is a Hamiltonian path from E to B in Figure 1 (a). We will consider two flavors of the Hamiltonian path problem. For the *given endpoint* Hamiltonian path problem we will be given two nodes a and b and determine if there is a Hamiltonian path between them. For the *any endpoint* Hamiltonian path problem we will determine if there is a Hamiltonian path between any two nodes in the graph.

Traveling salesman problem is to determine whether a traveling salesman can go through every city exactly once and return to the starting city, and the total traveling distance is within a given bound, when given the positive integer distance between every two cities. For example, A-B-D-E-C-A is a solution for the graph in Figure 1 (b) with a total distance 12.

Bounded degree spanning tree problem is to determine if a graph has a *bounded degree spanning tree* of a given degree D. A bounded degree spanning tree of degree D is a *spanning tree* in which the degrees of all tree nodes are bounded by D. A spanning tree is a tree that spans all nodes, i.e., all nodes are in the tree. For example, the thick edges in Figure 1 (c) is a spanning tree of bounded degree 3 because it covers all nodes and the maximum degree among all tree nodes is 3.

Please prove the following three statements. Your proof must be concise and address the key ideas only. **Each proof can only use at most *100* Chinese characters or *100* English words. *Over-sized proofs will be ignored.***

(a) (5 points) If the *given endpoint* Hamiltonian path problem is NP-complete, then the Hamiltonian cycle problem is also NP-complete.

(b) (5 points) If the Hamiltonian cycle problem is NP-complete, then the traveling salesman problem is also NP-complete.

(c) (5 points) If the *any endpoint* Hamiltonian path problem is NP-complete, then the bounded degree spanning tree problem is also NP-complete.

4. (20 points) Find a shortest path in a DAG with given constraints. A graph is a *DAG* if it is *directed* and it has *no* cycle. Every edge has a positive integer distance, and is either *thick* or *thin*. Figure 2 illustrates such a DAG. For example, the edge from A to B is thick and with a distance 1, and the edge from A to C is thin and has a distance 2.

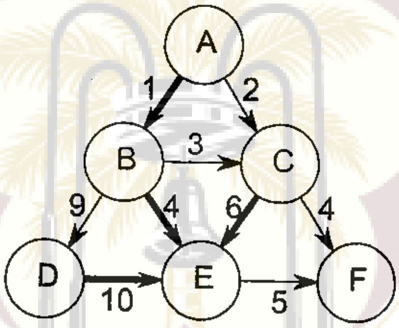

Figure 2: A DAG

Now given nodes A and F, and two integers k, n, find the shortest path from A to F that goes through exactly k thick edges and n thin edges, and does not go through any node twice. For example, the shortest distance from A to F that goes through two thick and two thin edges is 15 by going through A-B-C-E-F. The path A-C-E-F is not a valid solution because it only goes through one thick edge. The path A-B-D-E-F does go through the right number of edges, but its total distance is 25, which is not the minimum.

Please describe the key ideas in your algorithm *without using pseudo code.* Finally analyze the time complexity of your algorithm. We assume that there are N nodes and M edges in the DAG.

Your entire description and analysis can only use at most *400* Chinese characters or *400* English words. *Over-sized description or analysis will be ignored. Pseudo code will also be ignored.*

5. (30 points) Computational geometry has applications in a number of fields such as computer graphics, computer vision, robotics, molecular modeling and statistics. We consider the following problems of computational geometry in 2D space where the points are specified by their x and y coordinates, $p_i = (x_i, y_i)$.

接次頁

(a) (5 points) Give an algorithm to determine whether two line segments intersect. The inputs are segments $\overline{p_1p_2}$ and $\overline{p_3p_4}$, and the algorithm returns TRUE if the two segments intersect and FALSE if they do not.

(b) (10 points) Give an *efficient* algorithm to determine whether any two line segments in a set of segments intersect. Please describe the key ideas and discuss its time complexity.

(c) (15 points) Give an *efficient* algorithm to determine whether a given point is inside a given polygon. The vertices of the polygon are in counterclockwise order. We assume that the boundary of the polygon has no self-intersections, and that the given point is not on the polygon's boundary. Please describe the key ideas and discuss its time complexity.

Figure 3 illustrates a polygon. Point A is within the polygon and point B is outside of the polygon.

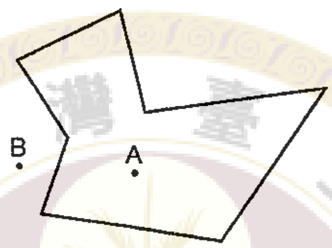

Figure 3: A Polygon and two points