1. (25 points) Given a source s and destination d pair of an undirected graph G=(V, E), write a C or C++ function including the data structure that finds all the disjoint paths from s to d.
2. (25 points) Consider a single-server priority system where an arriving customer is assigned a priority value p, p=1, 2, ..., P. The larger the class index, the higher the associated priority level. For customers of the same class, the queueing discipline is first-in-first-out. When a customer comes, if the server is busy he/she is queued in the buffer. While waiting in the buffer queue, an impatient customer may leave the system without service. Three major operations are identified for this system: getNextCustomerToServe(out customerInfo) operation which fetches an oldest customer with the highest priority class from the buffer queue; putCustomerIntoBuffer(in priorityClass, in customerInfo) operation which places a newly arrived customer into the buffer queue; and balkCustomer(in priorityClass, in customerInfo) operation which removes a specified customer from the buffer queue. Assuming array-based implementation of the buffer queue (with finite capacity), please design appropriate data structures for this system so that the time complexity of getNextCustomerToServe(), putCustomerIntoBuffer(), balkCustomer() operations are O(P), O(1) and O(N) respectively; and with minimum data movement. No data shifting in the buffer queue is allowed in any of the three operations. N is the number of customers in the system.

3. (20 points)

   (a) What are balanced search trees? Why are they useful? Please describe the structure of a particular kind of balanced search trees of your choice.

   (b) For the kind of balanced search trees you have described, please explain how a tree can be kept balanced when a new node is inserted into the tree. What is the cost of keeping the balance?

4. (20 points)

   (a) For an arbitrary weighted graph G, prove that, if the weights of all edges are distinct, then G has an unique minimum-cost spanning tree.

   (b) The celebrated Prim's algorithm for finding the minimum-cost spanning tree of a graph is based on the following property:

   > Consider a weighted graph G and an arbitrary subgraph G' of graph G. Let $E(G', G)$ be the set of edges connecting nodes of G' to nodes in G but not in G'. If $E(G', G)$ is not empty, then the edge with the minimum weight in $E(G', G)$ must be in the minimum-cost spanning tree of G.

   Prove the correctness of the above property.

接背面

5. (10 points) The Clique Problem and the Vertex Cover Problem can be described as follows.

   Clique: Given an undirected graph $G = (V, E)$ and an integer k, determine whether G contains a clique of size $\geq k$. (Note: A *clique* C is a subgraph of G such that all vertices in C are adjacent to all other vertices in C.)

   Vetex Cover: Given an undirected graph $G = (V, E)$ and an integer k, determine whether G has a vertex cover containing $\leq k$ vertices. (Note: A *vertex cover* of G is a set of vertices such that every edge in G is incident to at least one of these vertices.)

   Assume that we know the Clique Problem is NP-complete. Prove that the Vertex Cover Problem is also NP-complete.