國立交通大學 97 學年度碩士班考試入學試題

科目：資料結構與演算法(1001)　　考試日期：97年3月9日 第 1 節

系所班別：資訊系所跨組聯招　　組別：資訊聯招　　

＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

1. Please answer "True" or "False" for the following questions. Note that you will get -2 points as penalty if your answer is not correct. (10%)

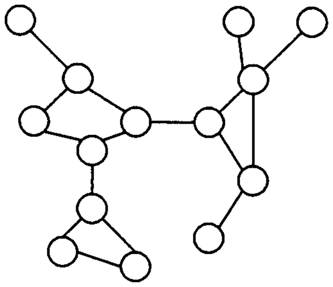
(1) The lower bound of worst case time complexity of sort algorithms is Ω(nlog n).

(2) The result of the postfix expression 23+4*6+ is 26.

(3) There are 101 threads in a threaded binary tree with 100 nodes.

(4) Every binary tree is uniquely defined by its pre-order and post-order sequences.

(5) There are 7 articulation points and 10 bi-connected components in the below graph.



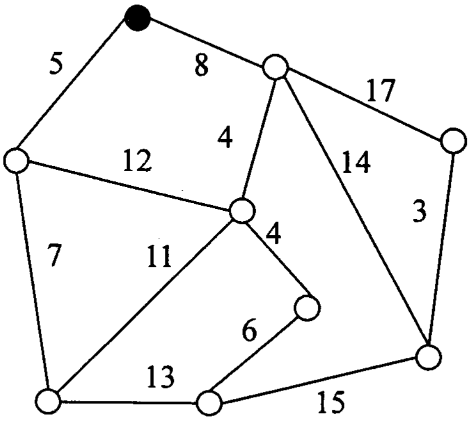
2. minimum spanning tree (10%)

(1) Please show the definition of minimum-spanning tree. (2%)

(2) Consider the graph below. Please use Kruskal's algorithm step-by-step to find the minimum-spanning tree. (3%)

(3) Consider the graph below. Let the gray node be the starting node. Please use Prim's algorithm step-by-step to find the minimum-spanning tree. (3%)

(4) If adjacency matrix is used, what are the time complexities of Kruskal's and Prim's algorithms? (2%)



＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

3. Heap (10%)

(1) Please show the definition of min heap (2%)

(2) Given a binary tree represented by the following structure, design an algorithm to check whether the binary tree is a min heap. Please describe the rationale of your algorithm first and then show your algorithm in C-like pseudo-code. (8%)

```
struct node
{
  int value;
  struct node * left; /* points to the root of the left subtree */
  struct node * right; /* points to the root of the right subtree */
};
```

4. Answer T(rue) or F(alse) to each statement. Each correct answer earns 3 points. There is a penalty, 1.5 points, for each wrong answer. 是非題，答對得 3 分，答錯扣 1.5 分。

(1)Consider a binary tree T. The maximum number of nodes on level k is 2^{k-1} (root is at level 1). The maximum number of nodes in T of level k is $2^k - 1$. Thus for any binary tree, T, has n nodes, the height is $\Omega(n \log n)$.

(2)To heapsort n integers stored in an array, we first build a maximum heap. Then we move the maximum (the one at root) to the end of the array and modify the heap of size decreased by one. Since the best algorithm to build the heap needs $O(n\log n)$ time, and since we have to modify the heap n times, and each iteration takes $O(\log n)$ time, heapsort can be done in $O(n\log n)$ time.

(3)Consider the operation to merge two priority queues. If we use a binary heap, the merge operation cannot be done efficiently. If we use a leftist tree data structure, the merge operation can be done in $O(\log n)$ time. Actually, algorithm for "delete minimum" and "insert arbitrary" into a leftist tree implemented priority queue is designed based on the merge operation of two leftist trees. Thus delete minimum and insert arbitrary can be done in $O(\log n)$ time.

(4)To argue any sorting algorithm needs at least $c \cdot n\log n$ operations, c is a constant, we argue that any sorting algorithm needs at least $c' \cdot n\log n$ comparisons, c' is a constant. Since "balance partition" achieves the best performance in divide and conquer approach. For example, if balance partition, merge sort can be done in $T(n) = 2T(n/2) + n$ time. Since $T(n)$ is $n\log n$, and that is the best performance, we need at least $c \cdot \log n$ comparisons and thus any sorting algorithm needs at least $\Omega(n\log n)$ operations.

(5)The least significant digit first radix-sort sorts n records in linear time. That is, we sort n records according to their key in the order from the least significant to the most significant; we then have the records sorted. In this algorithm, to sort according to each key, we need a stable sorting algorithm. Thus we can use merge sort or inserting sort algorithms. Quick sort or heap sort is not appropriate since they are not stable sorting algorithms.

(6)Set manipulation needs Union and Find operations. To achieve the most efficient implementation, a forest data structure is needed. In this case, Union can be done in constant time. Time required for Find depends on the height of the tree. If the weighting rule for Union and path collapsing rule for Find are applied, for n-1 Unions and m Finds, Each Find can be done in amortized $\alpha(m,n)$ time. $\alpha(m,n)$ is a function growing extremely slowly, it is almost constant (but not a constant).

(7)Balance tree supports efficient searching and dynamically insertion and deletion operations since we can guarantee that the height of a balance tree is $O(\log n)$. A balance tree can be a height-balance tree such as an AVL-tree or a 2-3-4 tree. It also can be a weight-balance tree. The weight-balance condition is usually determined by the number of leaves in the subtrees.

(8)Dijkstra's algorithm for single source shortest path problem needs an appropriate data structure to achieve the best performance. Since we have to determine the least weight vertex and the weight of vertex could be reduced by the so called "RELAX" operation, we need a binominal heap. In this case the Dijkstra's algorithm runs in $O(V\log V+E)$ time where V is the number of vertices and E is the number of edges in the graph.

5. Give tight asymptotic upper bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \le 2$. (You may just give the answer without any explanation) (10%)

(1) $T(n) = 4T(\frac{n}{4}) + n\log n$

(2) $T(n) = 4T(\frac{n}{4}) + \frac{n}{\log n}$

(3) $T(n) = 2T(\sqrt{n}) + \log n$

(4) $T(n) = T(n-1) + \frac{1}{n}$

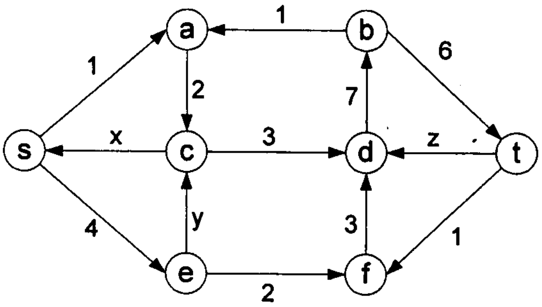
(5) $T(n) = T(\frac{n}{2}) + 1$

6. Consider the maximum flow problem in the following graph. Let s be the source node and t be the sink node. A flow in the network is indicated by the numbers on the edges.

(1) Assume that each edge has unlimited capacity. Determine the missing flows x, y, and z. (3%)

(2) Is it possible to reassign suitable edge capacities so that the flow indicated represents a maximum flow? Justify your answer. (2%)

(3) Now, assume the edge capacities are exactly the same as the flows shown in the graph. (For example, the number 1 on edge (s, a) represents that the flow is 1, and that the capacity is also 1. The flows x, y, and z are computed as in part (1).) Is the flow indicated a maximum flow? Then, find the value of a maximum flow, find the capacity of a minimum cut and indicate a minimum cut. (You may just give the answer without explanation.) (6%)



7. Let A[1],..., A[n] be a sequence of n positive real numbers. We want to re-arrange these numbers into B[1],..., B[n] such that

$(B[1] - B[2])^2 + (B[2] - B[3])^2 + \ldots + (B[n-1] - B[n])^2$ is maximized.

(1) Let n=8 and 1,2,3,4,5,6,7,8 be the sequence.
What is the maximum value for this input? (5%)

(2) Design an efficient algorithm for this problem.
What is the time complexity of your algorithm? (10%)

8. Given n positive integers, C[1],..., C[n], we can multiply each number by +1 or -1. In this problem we want to determine whether it is possible to multiply each number properly such that their sum is zero.

For example, let n=5, C[1]=1, C[2]=2, C[3]=3, C[4]=6, C[5]=4, then it is possible, because 1-2+3-6+4=0. On the other hand, for 1,2,3,4,5, it is impossible.

Now consider another problem, given a set S of m integers and an integer K, we want to determine whether there is a subset of S with the sum equal to K. This is the so called Subset Sum problem, which is a well known NP-complete problem. Prove that the above problem is NP-complete by reducing the Subset Sum problem to it. (10%)