CSL201 II Sem 2006-07, Major Exam

3:30PM to 5:30PM, 8th May 2007

Note. There are twenty multiple choice questions in this exam. Questions may have more than one correct answer. Marking **all** the correct answers of a given question gets you **1.5 marks**. Marking even a single wrong answer gets you **- 0.5 marks**. Marking some correct answers (and no wrong answers) gets you **0.5 marks**.

All answers are to be marked in the response sheet attached to the question paper. Please detach this response sheet from the question paper. There are 6 different question papers with codes from 1 to 6. **Please ensure that the code on top of your response sheet matches the code on the question paper.**

1. You are given a single stack and a sequence of six numbers 123456 that appear as input one at a time. As each number appears it can either be pushed on the stack or sent to output. When a number on the stack is popped it has to be sent to the output (there is no additional storage available.) Given this system, which of the following permutations of this sequence can be output?

   (a) 164235

   (b) 654123

   (c) 465321

   (d) 132546

2. Consider the 2-4 tree given in Figure 2. Which of the following statements are true:

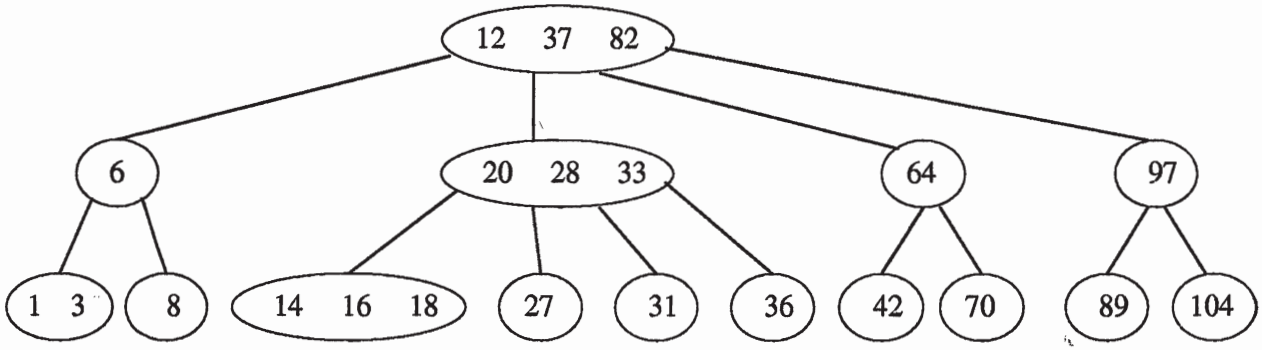

Figure 1: A 2-4 tree

   (a) Inserting 17 increases the height of the tree.

   (b) The tree could have 28 in its root node after 17 is inserted.

   (c) Inserting 90, 91, 92, 93, 94, 95, 96 increases the height of the tree.

   (d) Inserting 29, 30, 32, 33 decreases the height of the tree.

3. Suppose we are given a minimum spanning tree T for a graph G. Let $e = (u, v)$ be an edge of G which is not in T. Suppose we **decrease** the length of edge e from l_e to l'_e. In which of the following cases, we can be **sure** that T will **not** remain a minimum spanning tree.

   (a) l'_e is less than the lengths of all other edges which have an end-point in u or v.

   (b) The path from u to v in T has an edge of length larger than l'_e.

(c) There is a cycle in G which contains e such that l'_e is the length of the smallest edge in this cycle.

(d) Every cycle containing e has at least one edge of length larger than l'_e.

4. A k-heap is a heap where each non-leaf node has exactly k children. It has the same properties as those of a binary heap. What is the worst case running time of **insert** and **delete-min** operations (let n denote the number of elements in the heap).

   (a) $O(\log_k n)$ and $O(k \log_k n)$.

   (b) $O(k \log_k n)$ and $O(\log_k n)$.

   (c) $O(k \log_k n)$ and $O(k \log_k n)$.

   (d) $O(\log_k n)$ and $O(\log_k n)$.

5. The inorder traversal for a binary tree is DEACBFGIKJOLPHMNRQS. The preorder and postorder traversals are:

   (a) IGFEDCABHJKLOPMNQRS and DABCEFGKOPLJRSQNMHI

   (b) IGFEDCBAHJKLOPMNQRS and DBACEFGKOPLJRSQNMHI

   (c) KIGFEDCABHJLPOMNQRS and DABCEFGIPOLJRSQNMHK

   (d) KIGFEDCABHJLOPMNQRS and DABCEFGIOPLJRSQNMHK

6. Suppose that while your computer is sorting an array of objects, its memory is struck by a cosmic ray that changes exactly one of the keys to something completely different.For which of these sorting algorithms will the final array have just one or two keys out of place in the worst-case scenario ?

   (a) Quick sort

   (b) Merge sort

   (c) Radix sort

   (d) Heap sort

7. Suppose that we are using double hashing and have selected the table size m to be 1200. When we do an insertion, the first hash function determines the point at which we initially probe the table, and the second hash function determines the amount by which we jump (mod 1200) as we search for an empty table position. Which of the values below for the second hash function would guarantee that we will find an empty position if there is one in the table?

   (a) 95

   (b) 74

   (c) 53

   (d) 39

   (e) 27

8. Arrange the following functions in order of their asymptotic grwoth rates, i.e., if $f(n)$ appears to the left of $g(n)$, then $f(n)$ is $O(g(n))$:

$$2^n, n!, n^n, n^{\log^3 n}$$

   (a) $n^{\log^3 n}, 2^n, n^n, n!$

   (b) $n^{\log^3 n}, 2^n, n!, n^n$

(c) $2^n, n^{\log^3 n}, n^n, n!$

(d) $2^n, n^{\log^3 n}, n!, n^n$

9. Suppose we start with an AVL tree T . We do an insertion of a new key in the tree just using the binary search tree algorithm without yet doing any rotations; the result is a tree T'. It turns out that the deepest node in the tree with a balance outside of $\{1, 0, -1\}$ is is the root, so we do an appropriate rotation at the root to produce the final AVL tree T''. If the height of T'' is 10, what were the heights of T and T' respectively ?

   (a) 10 and 11

   (b) 11 and 10

   (c) 11 and 11

   (d) 10 and 10

   (e) We do not have enough information to determine the answer.

10. Which of the following sorting algorithms cannot be implemented as a stable sorting algorithm.

    (a) Quick sort

    (b) Merge sort

    (c) Radix sort

    (d) Heap sort

11. Consider an undirected connected graph G with edge lengths. Assume all edge lengths are positive integers. We would like to find the **length** of shortest path from s to every other vertex in G. We initialize an array $D[\cdot]$ to $D[s] = 0$, $D[v] = \infty$ if $v \neq s$. Consider the following algorithm. Here $l(e)$ denotes the length of edge e.

```
repeat
   find an edge e = (u,v) such that D[u] > D[v] + l(e)
   update D[u] to D[v] + l(e).
until no such edge can be found
```

    Which of the following statements are true about this algorithm.

    (a) The algorithm may not terminate on some graphs.

    (b) The algorithm always terminates. But there exist graphs for which the final values $D[v]$ are not equal to the length of shortest path from s to v for some vertices.

    (c) The algorithm always terminates. Further, on termination the values $D[v]$ are equal to the length of the shortest s to v path for every vertex v.

    (d) The values $D[v]$ are always at least the length of the shortest s to v path.

12. Suppose we want to find shortest path from a source vertex s to a vertex t in an undirected graphs. However some edges in the graph may have negative lengths. In which of the following cases will the Dijkstra's algorithm find the desired shortest path ?

    (a) No shortest path from s to t has an edge of negative length.

    (b) There is at most one edge of negative length

(c) There does not exist a cycle in which the sum of the lengths of the edges in it is negative.

(d) None of the above.

13. Suppose that we have built up a compressed trie for a list of strings. The rightmost path of this trie is as shown in Figure 2, where the numbers to the left of each node specify the total length of the labels on the path from the root to that node. Let v be the lexicographically largest of the strings we have inserted. We now add a new string w which is lexicographically larger than v, and for which the length of the longest common prefix of v and w is 17. In the new compressed trie, how many edges are there on the rightmost path?

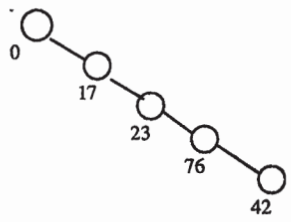

Figure 2:

(a) 5

(b) 4

(c) 3

(d) 2

(e) 1

14. Consider again the 2-4 tree of height 2 given in Figure 2. Which of the following statements are true:

(a) After 27 is deleted the final tree has 20 a leaf node.

(b) After 37 is deleted the final tree has 42 in the root node.

(c) 7 is the min number of keys in a 2-4 tree of height 2.

(d) At least 15 keys have to be deleted from the tree in Figure 2 to decrease its height.

15. Suppose we run the DFS and the BFS algorithms on a connected undirected graph G. It so happens that the DFS tree and the BFS tree obtained from these algorithms are identical. What can we claim about the graph ?

(a) The graph does not have a cycle

(b) The graph can have one cycle

(c) The graph can have several cycles but no two of these cycles share a common vertex.

(d) No such conclusion can be drawn.

16. In a skip list each node is promoted to the next level with probability $1 - \frac{1}{\log n}$. The expected height of a node in such a skip list is

(a) $O(1)$

(c) $\theta(\log n)$

(d) $\theta(1)$

(d) $O(\log \log n)$

17. Consider a connected undirected graph G. Each edge e has **two** lengths, namely $l(e)$ and $w(e)$. Let s be a starting vertex in G. Consider the following modified Dijkstra's algorithm. Here $D[\cdot]$ and $F[\cdot]$ are two arrays initialized to $D[s] = F[s] = 0$ and $D[v] = F[v] = \infty$ if $v \neq s$. Let n denote the number of vertices in G.

```
Initialize a set M to emptyset
While M has less than n elements do
   pick a vertex v which is not in M and for which D[v] is minimum
   add v to M
   for every neighbour u of v such  that u is not in M do {
         if D[u] > D[v] + l((u,v)) {
            update D[u] = D[v} + l((u,v))
                   F[u] = F[v] + w((u,v))
            update parent[u] = v

          }
         else if D[u] == D[v] + l((u,v)) AND F[u] < F[v] + w((u,v)) {
            update F[u] = F[v] + w((u,v))
            update parent[u] = v
         }
   }
```

What does this algorithm do ? For a path P, and a function $f(e)$ on edges of the graph, define $f(P)$ as the sum of $f(e)$ for all edges e in P.

(a) For every vertex v, it finds a path P from s to v for which $f(P)$ is minimum, where $f(e) = l(e) + w(e)$.

(b) For every vertex v, it finds a path P from s to v for which $f(P)$ is minimum, where $f(e) = \min(l(e), w(e))$.

(c) For every vertex v, it finds a path from s to v for which $l(P)$ is minimum. Further if there are several shortest paths from s to v in terms of length $l(e)$, then it outputs the path among these for which $w(P)$ is minimum.

(d) For every vertex v, it finds a path from s to v for which $w(P)$ is minimum. Further if there are several shortest paths from s to v in terms of length $w(e)$, then it outputs the path among these for which $l(P)$ is minimum.

18. Consider a linked list where each node has a **next** pointer. We say that the list has a loop if there are nodes $a_1, a_2, \ldots, a_k$ in the list such that the a_1.next $= a_2, a_2$.next $= a_3, \ldots, a_{k-1}$.next $= a_k$ and a_k.next $= a_1$. Consider the following code to find out if a list L has a loop or not. Here p and q are pointers. What is the worst case running time of this code ? Let n denote the number of nodes.

```
p =  L
q = p.next
loop_exists = FALSE
```

```
while (p!=NULL && loop_exists = FALSE) {
    if (p == q)
         loop_exists = TRUE
    else
        { p = p.next
          q = (q.next).next
        }
}

return loop_exists
```

(a) $O(n^2)$

(b) $O(\log n)$

(c) $O(n \log n)$

(d) $O(n)$

19. Suppose an undirected graph G has 100 vertices and 28 edges. What are the maximum and the minimum number of possible connected components in the graph ?

(a) 93 and 86

(b) 93 and 72

(c) 86 and 72

(d) 93 and 86

20. Define the length of a cycle as the number of edges (or vertices) in the cycle. The **girth** of a graph is the smallest length of a cycle in the graph. Consider the following algorithm find the girth of a connected undirected graph.

```
Pick a vertex u in G.
Starting from u, run the BFS algorithm on G.
Let T be the BFS tree obtained
Define the level of a vertex v as its level in this tree (level of root is 0)
For every edge e=(v,w) which is not in T
   compute a quantity Q(e) = 1 + level(v) + level(w).
Output the smallest Q(e).
```

Which of the following are true about this algorithm ?

(a) The algorithm outputs the girth of the graph.

(b) The algorithm outputs the girth if the starting vertex u lies in a cycle which has length equal to the girth of the graph. Otherwise the graph may give a wrong result.

(c) The algorithm will always give a wrong result if the starting vertex u does not lie on a cycle whose length is equal to the girth of the graph.

(d) There are graphs for which the algorithm will give a wrong answer no matter how we pick the starting vertex u.