VANCOUVER ISLAND UNIVERSITY
CSCI 260 — MIDTERM EXAMINATION
18 October 2018, 11:30 — 12:50

Duration: 80 Minutes **Instructors:** H. Liu

TO BE ANSWERED IN BOOKLETS

Instructions

- Students must count the number of pages in this examination paper before beginning to write, and report any discrepancy immediately to the invigilator.
- This examination paper consists of 5 pages.
- This is a CLOSED BOOK examination. You are allowed to bring one piece of letter-sized and double-sided notes.
- Calculators are NOT permitted.
- Remember to state any assumptions and show rough work.
- Note carefully the weight of each question, and answer appropriately.
- Attempt all questions. All questions relate to material covered in the lectures, labs and assignments.

Reference

The Master Theorem: Let $f(n)$ and $T(n)$ be defined as:

$$T(n) = \begin{cases} c & \text{if } n < d \\ aT(n/b) + f(n) & \text{if } n \geq d. \end{cases}$$

where $d \leq 1$ is an integer constant, $a > 0, c > 0$, and $b > 1$ are real constants and $f(n)$ is a function that is positive for $n \geq d$.

1. If there is a small constant $\varepsilon > 0$, such that $f(n)$ is $O(n^{log_b a - \varepsilon})$, then $T(n)$ is $\Theta(n^{log_b a})$.

2. If there is a constant $k \geq 0$, such that $f(n)$ is $\Theta(n^{log_b a} log^k n)$, then $T(n)$ is $\Theta(n^{log_b a} log^{k+1} n)$.

3. If there are small constants $\varepsilon > 0$ and $\delta < 1$, such that $f(n)$ is $\Omega(n^{log_b a + \varepsilon})$ and $af(n/b) \leq \delta f(n)$, then $T(n)$ is $\Theta(f(n))$.

1. (8 Marks) Given an array of size N, each element in the array has the following struct data type:

```
struct Data {
    int priority; // the smaller the number, the higher the priority
    string content;
};
```

Write a c++ function to illustrate an algorithm that checks whether all the elements in the array form a heap.

```
bool isHeap(Data A[], int N);
```

2. (5 Marks) Each node in a binary tree has the data type `Node` as defined below:

```
struct Node {
    Data *dataptr;
    Node *left;
    Node *right;
};
```

Write a C++ function that performs the AVL tree's single left rotation on a critical node (passed into the function as a parameter).

3. (10 Marks)

   (a) List the main operations of a priority queue ADT. For each operation, describe its required parameters and its functionality.

   (b) Suppose that an unsorted array is used to implement the priority queue, write a C++ function to illustrate the algorithm that implements the operation of removing the object with the highest priority.

4. (5 Marks) Is it true that the best performance of all sorting algorithms is O(NlogN), where N represents the number of elements to be sorted and may be a very large number. Justify your answer clearly and briefly. You can use examples in your justification.

5. (8 Marks) The private data fields of a class `HashTable` are shown below:

```
class HashTable {
  public:
    ...
  private:
    enum Status {Empty, Occupied, Removed};
    int h(int k); // hash function
    struct Cell {
        int key;
        string element;
        Status status;
    };
    Cell data[79];
    int size;
};
```

And the method `insert` of the `HashTable` class is shown below:

```
void HashTable::insert(int k, string e)
{
   if (size >= 79)
      throw "Hash Table Full";

   int position = h(k);
   while (data[position].status == Occupied)
       position = (position + 1) % 79;

   data[position].key = k;
   data[position].element = e;
   data[position].status = Occupied;
   size++;
}
```

Write a C++ function that implements the method `lookup` of the `HashTable` class.

6. (10 Marks) On an input of size N, when $N > 1$, an algorithm solves a problem recursively by solving three (3) sub-problems each of the size $\lceil N/3 \rceil$, and then "combining" them to form the final result in a further $NlogN$ steps. A problem of size 1 takes 5 steps to solve in this algorithm. The full runtime of the algorithms is reflected by the number of steps required.

   (a) Present the recurrence equation T(N) for the runtime of the above described algorithm.

   (b) Solve this recurrence equation, i.e., find a non-recurrence function g(N) such that $T(N) \in \Theta(g(N))$.

7. (4 Marks) Assuming that a set of integer numbers are stored in a general **binary** tree, one number in each node of the tree. When the tree is traversed, the numbers are printed out in the following order in each traversal:

   - In pre-order traversal: 5, 1, 8, 6, 2, 3, 9, 4, 7
   - In post-order traversal: 6, 2, 8, 1, 4, 9, 7, 3, 5
   - In in-order traversal: 1, 6, 8, 2, 5, 9, 4, 3, 7

   Draw a general binary tree that is consistent with the above provided information. Your tree should clearly show the integer numbers stored in each node. (Note that the values of the numbers don't have any significant meanings.)

====== END OF EXAM QUESTIONS ======