National University of Computer and Emerging Sciences, Lahore

CS210 **Data Structures and Algorithms**

Final Exam, Wed, 28 Dec 2011, Time: 3 hr, Maximum Marks: 70

Handwriting must be legible. Clearly label figures. Clearly state assumptions made, if any. Code must be adequately commented. There are 7 questions in this question paper.

---

1. Solve the following recurrence relation.

$$T(n) = \begin{cases} 4\,T(n/2) - 1 & n > 4 \\ 5 & 0 < n \leq 4 \end{cases}$$

   You may find the following equation helpful in solving this recurrence relation.[10]

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}$$

2. Write a function to merge two sorted linked-lists. Assume that the lists are already sorted in non-decreasing order. Function prototype is given below.

```
struct cell_ *merge_lists (struct cell_ *l1, struct cell_ *l2);
```

   where the structure of each cell of both linked lists is defined as follows.

```
struct cell_ {
  int data;
  struct cell_ *link;
};
```

   The merged list must also be in non-decreasing order. Your code must not traverse each of the two lists more than once, and must not allocate any additional memory. *Write line numbers next to your code and explain it briefly using these numbers.*[10]

3. Consider the following set of 10 integers $\{35, 8, 1, 4, 18, 25, 12, 83, 63, 17\}$. Find a sequence insertions of these numbers which when applied to an empty binary search tree results in tree of the form shown in Figure 1. Briefly explain how you found the sequence and if this is the only sequence that forms a tree of this shape.[10]

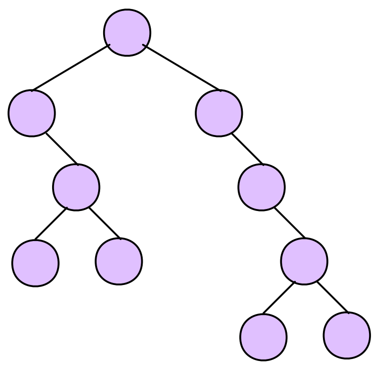

Figure 1: Binary Tree.

4. Sort the following array of integers by Radix Sort algorithm. Show all steps of the execution of the algorithm.[10]

   7643, 1352, 4362, 3, 5436, 346, 463, 9003, 2814, 46

5. A doubly linked is defined in `dlist.h` header file as shown in Listing 1.

   Listing 1: `dlist.h`

```c
/* various header files */

typedef struct dlist_ *dlist_adt;

/* function prototypes */
dlist_adt dlist_init (void);
int dlist_insert (dlist_adt l, int n, int loc);
/* <other functions of dlist ADT> */
```

   A partial implementation file is shown in Listing 2. Implement `dlist_init` and `dlist_insert` functions.[10]

   Listing 2: `dlist.c`

```c
/* other headers files */
#include "dlist.h"

struct cell_ {
   int data;
   struct cell_ *flink;  /* pointer to next cell     */
   struct cell_ *blink;  /* pointer to previous cell */
};

struct dlist_ {
   struct cell_ *head;
   struct cell_ *tail;
   int count;
};

struct dlist_ *dlist_init (void)
{
   /* <implement this function> */
}

int dlist_insert (dlist_adt l, int n, int loc)
{
   /* n is the number to be entered in the list. */
   /* loc is the location of n after insertion.  */
   /* assume first cell is at location 1.        */
   /* <implement this function> */
}
```

6. Dr. A. Apple Banana is professor of Computer Science at Cucumber University. He has produced a scholarly piece of code to be published in a respected journal. The code is re-produced in Listing 3. Please determine what the `banana` function is doing.[10]

Listing 3: Dr. Banana's Code

```
double apple (double carrot, int lemon)
{
   double garlic = 1;

   for ( ; lemon ; --lemon) {
      garlic *= (carrot / lemon);
   }
   return (garlic);
}

double banana (double carrot, int lemon)
{
   if (!lemon) {
      return (1);
   }
   return (apple(carrot, lemon) + banana(carrot, lemon - 1));
}
```

7. Determine whether $f(n) = O(g)$ for each of the following pairs of functions. A function $f(n)$ is said to be in the set $O(g)$ if $f(n) \leq g(n), \forall n \geq n_0$, where $c > 0$ is a constant.[10]

   (a) $f(n) = (n^2 - n)/2, \quad g(n) = 6n$

   (b) $f(n) = n + 2\sqrt{n}, \quad g(n) = n^2$

   (c) $f(n) = n + n \log_2 n, \quad g(n) = n\sqrt{n}$

   (d) $f(n) = n^2 + 3n + 4, \quad g(n) = n^3$

   (e) $f(n) = n \log_2 n, \quad g(n) = n\sqrt{n}/2$

Good Luck!