

# EE 453/717: Advanced Computing for Electrical Engineers

Instructors: Saravanan Vijayakumaran, Sachin Patkar

Indian Institute of Technology Bombay

Autumn 2011

Midsemester Exam : **25 points**

**Duration:** 120 minutes

---

Each of the following questions is worth 5 points.

1. Suppose we are given an image with a single square component in it and we are given the location of a starting pixel in the interior of the square. For example, consider the following image where there is a  $4 \times 4$  square component of ones and the underlined one represents the starting pixel.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & \underline{1} & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Write a C++ function which outputs the locations of the four corners of the square component in clockwise order starting from the top left corner. Assume the image has already been stored in a two dimensional array of integers `int **grid` and a pixel position is described by the structure `struct Position{int row, col;}`. The signature of the function should be `void FindCorners(int **grid, Position startPixel)` and the output should be pairs of integers corresponding to the locations of the corners. For the above example, the output should be following.

Top Left Corner: 3 3

Top Right Corner: 3 6

Bottom Right Corner: 6 6

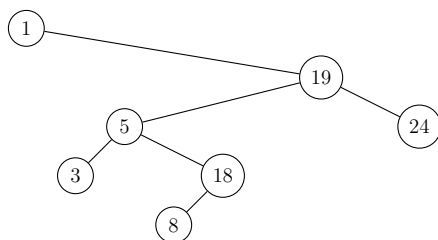
Bottom Left Corner: 6 3

2. Suppose a polynomial with real coefficients is represented as a linked list of its terms. Each node in the linked list represents a term and is defined in the following manner.

```
struct term
{
    int degree;    // Degree of the term (always nonnegative)
    double coeff; // Coefficient of the term
    term* next;
};
```

The nodes are ordered in decreasing order of the degrees i.e. the first node contains the term with highest degree. The zero polynomial is represented by a linked list containing one node which has both `degree` and `coeff` equal to zero. For all nonzero polynomials only terms with nonzero coefficients are present in the linked list. Write a C++ function `term* AddPoly(term* poly1, term* poly2)` which takes two linked lists representing polynomials as input and returns a linked list which represents the sum of the polynomials. Make sure that the sum of the polynomials does not contain terms with zero coefficients.

3. Draw the following binary search tree (BST) after the number 7 has been inserted such that it appears as a leaf node. Move the number 7 to the root of the BST using tree rotations. Draw the BST after **each** rotation operation (five rotations will be needed).



4. Consider a max heap of integers which is implemented using pointers. Each node in the max heap contains pointers to its parent node, its left child and its right child.

```

struct node
{
    int value;
    node* leftChild;
    node* rightChild;
    node* parent;
}
  
```

Write a C++ function which inserts an integer into this max heap such that the heap condition is satisfied. Assume that the current size of the heap is passed to the function as an argument. The signature of the function is `void Insert(node* root, int x, int N)` where `root` is a pointer to the root node in the heap, `x` is the integer which is to be inserted and `N` is the current size of the heap. *Hint:* If we number the nodes in the heap left to right and top to bottom starting from 1, the number of the location where the new node is inserted is  $N+1$ . Consider the binary representation of  $N+1$  without the leading 1 i.e. if  $N+1 = 6$ , the binary representation of 6 without the leading 1 is 10. For  $N+1 = 4$ , the binary representation of 4 without the leading 1 is 00.

5. Write down the output of the following program.

```

#include <iostream>
using namespace std;
  
```

```

#define N 9

template <class Item>
void exch(Item &A, Item &B)
{ Item t = A; A = B; B = t; }

template <class Item>
void foo(Item a[], int l, int r)
{
    if (r <= l) return;
    int i = bar(a, l, r);
    foo(a, l, i-1);
    foo(a, i+1, r);
}

template <class Item>
int bar(Item a[], int l, int r)
{
    int i = r+1, j = l; Item v = a[l];
    for (;;)
    {
        while (a[--i] > v) ;
        while (v > a[++j]) if (j == r) break;
        if (j >= i) break;
        exch(a[i], a[j]);
    }
    exch(a[i], a[l]);
    for (int j = 0; j < N; j++)
    {
        cout << a[j] << " ";
    }
    cout << endl;
    return i;
}

int main(int argc, char *argv[])
{
    char a[] = {'A', 'N', 'E', 'X', 'A', 'M', 'P', 'L', 'E'};

    for (int i = 0; i < N; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;
}

```

```
foo(a, 0, N-1);

for (int i = 0; i < N; i++)
{
    cout << a[i] << " ";
}
cout << endl;

}
```