Tutorial-2
IIIT-DELHI
Instructor: Debarka Sengupta

Question 1

a)Predict the output of the following program. What does the following fun() do in general?

```
#include<stdio.h>
int fun(int a, int b)
{
   if (b == 0)
       return 0;
   if (b % 2 == 0)
       return fun(a+a, b/2);
   return fun(a+a, b/2) + a;
}

int main()
{
  printf("%d", fun(4, 3));
  getchar();
  return 0;
}
```

b) In question 1, if we replace + with * and replace return 0 with return 1, then what does the changed function do? Following is the changed function.

```
#include<stdio.h>
int fun(int a, int b)
{
   if (b == 0)
       return 1;
   if (b % 2 == 0)
       return fun(a*a, b/2);
     return fun(a*a, b/2)*a;
}
```

```
int main()
{
  printf("%d", fun(4, 3));
  getchar();
  return 0;
}
```

Question 2

Explain the functionality of the following function

a)

```
int fun1(int x, int y)
{
  if(x == 0)
    return y;
  else
    return fun1(x - 1,  x + y);
}
```

b)

```
void fun(int x)
{
  if(x > 0)
  {
     fun(--x);
     printf("%d\t", x);
     fun(--x);
  }
}

int main()
{
  int a = 4;
  fun(a);
  getchar();
  return 0;
}
```

c)

```
int fun(int n)
 {
   if (n > 100)
     return n - 10;
   return fun(fun(n+11));
 }

int main()
{
  printf(" %d ", fun(99));
  getchar();
  return 0;
}
```

d)

```
int fun(int count)
{
    printf("%d\n", count);
    if(count < 3)
    {
      fun(fun(fun(++count)));
    }
    return count;
}

int main()
{
    fun(1);
    return 0;
}
```

Question 3

Consider the following Java method (assume that it will only be called with 0 ≤ k and k ≤ n):

```
int choose (int n, int k) {
        if ((k == 0) || (k == n)) return 1;
        return choose (n-1, k) + choose (n-1, k-1);}
```

Explain why the running time will not be polynomial in n in general. Describe (in code or in English) how to revise the algorithm to be polynomial time in n. Carefully determine and justify a big-O bound on the running time of your improved version.

Question 4

True or false with justification: Let k be any positive integer constant greater than 1 . Then the recurrence T(n) = kT(n/k) + O(n), with T(1) = O(1), has the same big-O solution for any such k. (You may assume that T(n) is evaluated only when n is a power of k.)

Question 5

Consider the following algorithm strangeSort, which sorts n Comparable items in a list A. assumption: the n items are all distinct. Otherwise, the algorithm below fails to terminate if given a list of two or more items that are all equal.:

1) If n ≤ 1, return A unchanged
2) For each item x in A, scan A and count how many other items in A are less than x
3) Put the items with counts less than n/2 in a list B
4) Put the other items in a list C
5) Recursively sort B and C using strangeSort
6) Append the sorted C to the sorted B and return the result

a)Prove by induction on n that strangeSort correctly sorts all lists of length n, with smaller items first.

b) Formulate a recurrence for the running time T(n) of strangeSort on an input list of size n. Solve this recurrence to get the best possible big-O bound on T(n) -- you may assume if you like that n is a power of 2.

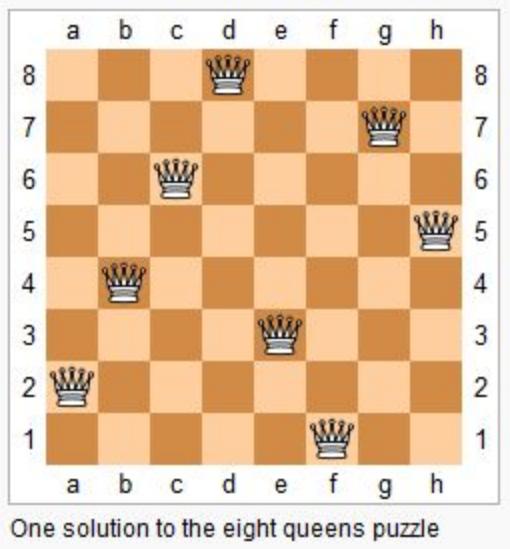
Question 6

Given s1, s2, s3, find whether s3 is formed by the interleaving of s1 and s2.
Example,
Given:
s1 = "aabcc",
s2 = "dbbca",

When s3 = "aadbbcbcac", return true.
When s3 = "aadbbbaccc", return false.

Question 7

The n-queens puzzle is the problem of placing n queens on an n×n chessboard such that no two queens attack each other.



One solution to the eight queens puzzle

Given an integer n, return all distinct solutions to the n-queens puzzle.
Each solution contains a distinct board configuration of the n-queens' placement, where 'Q' and '.'both indicate a queen and an empty space respectively.
For example,
There exist two distinct solutions to the 4-queens puzzle:

```
[
 [".Q..",  // Solution 1
  "...Q",
  "Q...",
```

```
  "..Q."],

 ["..Q.",  // Solution 2
  "Q...",
  "...Q",
  ".Q.."]
]
```

Question 8

Given a string s, partition s such that every string of the partition is a palindrome.
Return all possible palindrome partitioning of s.
For example, given s = "aab",
Return

```
 [
   ["a","a","b"]
   ["aa","b"],
  ]
```

Question 9:

The set [1,2,3,…,n] contains a total of n! unique permutations.
By listing and labeling all of the permutations in order,
We get the following sequence (ie, for n = 3) :

1. "123"
2. "132"
3. "213"
4. "231"
5. "312"
6. "321"

Given n and k, return the kth permutation sequence.
For example, given n = 3, k = 4, ans = "231"