Cairo University
Faculty of Engineering
Computer Engineering
Fall 2019

Design and Analysis of Algorithms

Sheet 1 – Recursion, Brute-force, and Alpha-beta Pruning

1. [Recursion] Write a recursive function to print all 2^n value for a given n. Example: given $n = 3$, print “000”, “001”, “010”,…,”111”.
    a. Trace your code and draw the recursion tree for $n = 3$. In each node of the recursion tree, state the values of all variables
    b. Draw the contents of the variables placed in the stack and the heap during the execution of the function till printing the first two values; “000” and “001”.

2. [Recursion] Write a recursive function to print all k^n value for a given radix k and n. Example: given $k = 3$ and $n = 4$, print “000”, “001”, “002”,”010”…,”222”.

3. [Brute-force] Given an array of integers, find the length of the Longest Increasing Subsequence (LIS) inside the array such that all elements of the subsequence are sorted in increasing order. For example, the length of LIS for {12, 24, 9, 35, 21, 50, 41, 62, 82} is 6 and LIS is {12, 24, 35, 50, 62, 82}. Hint: think of it as having ‘n’ values in the list and you are trying all subsets of the list similar to trying all 2^n subsets of this list. Refer to problem 1.

4. [Brute-force] A child is running up a staircase with n steps and can hop either 1 step, 2 steps, or 3 steps at a time. Implement a method to count how many possible ways the child can run up the stairs.

5. [Brute-force] The graph coloring problem is an assignment of a color to each node of a graph with the constraint that no two connected/adjacent nodes have the same color while using the minimum number of colors. Write a program to find the minimum number of colors needed for a given graph.

6. [Brute-force] Given an $n \times n$ matrix with each cell in the matrix has the cost of passing through the cell. Moving from one cell to another can be in the four basic directions only (i.e., no diagonal movement allowed). Find the path from the top left cell to bottom right cell with minimal cost.

   In the shown example, the optimal path is 3-1-3-2-2.

3	1	3
8	16	2
2	2	2

7. [Alpha-beta Pruning] Solve again the above problem but with limiting the search tree early enough when the visited cells of a path is clearly not optimal (i.e., when the cost incurred at any point is already worse than the best path found so far).

8. [Brute-force] A 3-sat problem is the ANDing of Boolean expressions where each expression is the ORing of three literals/variables. The answer for this problem is YES or NO; whether there is a Boolean assignment for each variable such that all clauses are satisfied or not. It is enough for every clause to have one satisfying literal and the other two remaining have arbitrary values. Write a program to try all possible values to determine whether a given instance is satisfiable or not.
   Example: $(\bar{a} \vee b \vee \bar{c}) \wedge (\bar{a} \vee b \vee c) \wedge (a \vee \bar{b} \vee c)$