Discrete Mathematics (SC612)
Tutorial 4
29th Oct0ber, 2021

1. Consider a grid of squares, where each square is to be coloured with one of two colours black or white.

   (a) How many colourings are there for a 2×2 grid.

   (b) How big a grid do you need of the type $2 \times n$ so that every possible configuration of colours of a 2×2 grid occurs at least once?

   (c) What is the largest grid of the type $2 \times n$ such that there is no repeat of the same colour configuration 2×2 grid?

2. (a) Suppose the integers 1 to 10 are randomly written around a circular wheel, show that there will be three successive positions such that the sum of the numbers written there is at least 17.

   (b) If you were to generalise this to integers 1 to n around a wheel, what number would replace 17 as a guaranteed lower bound on maximum value of total of three successive positions?

3. Suppose for positive integers a, b, c, we are told that $a+b+c+ab+bc+ac$ is odd. Is $a+b+c$ even or odd or can take either value?

4. Consider the following table:

Key Value	8	1	7	3	2	4	5	6
Search Frequency	13	94	59	38	87	22	17	60

   We have already seen binary search trees arranged to respect the binary search tree property on key values. When we alter the tree structure the key value placements get changed and based on insertion order we get different trees. In this instance, you are required to construct the binary search tree on the key values (row 1) such that the product of the search frequency of a node with the distance of that node from the root totalled over all nodes is as small as possible.

From the stand-point of the data structure, the search frequency represents how often a particular key is searched for. Thus, we naturally want frequently searched terms to be closer to the root as it will take less time for the search.

Students may also try and devise a general algorithm for this problem given an arbitrary table of the type given in this instance. It involves the algorithm design technique or paradigm called **dynamic programming**.

solution:

Total subtree weight	1	2	3	4	5	6	7	8
1	94	181	219	241	258	318	377	390
2		87	125	147	164	224	283	296
3			38	60	77	137	196	209
4				22	39	99	158	171
5					17	77	136	149
6						60	119	132
7							59	72
8								13

The entry (i, j) of this table gives the total cost of the nodes of the key values from i to j.

To get the optimal solution for any subtree of the range (i, j), we need to minimise the answer

$$OPT_{i,j} = \min_{t=i}^{j}\{OPT_{i,t-1} + wt_{i,t-1} + OPT_{t+1,j} + wt_{t+1,j}\}$$

(a) Optimal cost of single node trees is 0

(b) Cost of two node subtrees is the weight of the smaller node. Thus $OPT_{1,2} = 87, OPT_{2,3} = 38, OPT_{3,4} = 22, OPT_{4,5} = 17, OPT_{5,6} = 17, OPT_{6,7} = 59, OPT_{7,8} = 13$

(c) Now for subtrees of three nodes:

- $OPT_{1,3} = \min\{OPT_{1,2}+wt_{1,2}, OPT_1+wt_1+OPT_3+wt_3, OPT_{2,3}+wt_{2,3}\} = \min\{87 + 181, 0 + 94 + 0 + 38, 38 + 125\} = 132$
- $OPT_{2,4} = \min\{OPT_{2,3}+wt_{2,3}, OPT_2+wt_2+OPT_4+wt_4, OPT_{3,4}+wt_{3,4}\} = \min\{38 + 125, 0 + 87 + 0 + 22, 22 + 60\} = 88$
- $OPT_{3,5} = \min\{OPT_{3,4}+wt_{3,4}, OPT_3+wt_3+OPT_5+wt_5, OPT_{4,5}+wt_{4,5}\} = \min\{22 + 60, 0 + 38 + 0 + 17, 17 + 39\} = 55$
- $OPT_{4,6} = \min\{OPT + wt, OPT_+ wt_+ OPT_+ wt_, OPT + wt\}$
- $OPT_{5,7} = \min\{\}$
- $OPT_{6,8} = \min\{\}$