Q2 Report
170050068, 170050081, 170050083, 170050100

Algorithm:
Find the bias of all possible valid paths(subgraphs) and get the path with maximum bias.

Used a backtracking algorithm to go through all the paths in the SPN. Also used memoization to improve the performance as there are many repetitive sub structures in the problem.

Path with maximum bias is stored as a list and updated whenever we encounter a different path with greater bias.

Greedy Strategy Validity:

A greedy strategy will not work in this problem for reasons similar to that of the Minimum Cost Path problem.

Let's take an example of two stage SPN.
With g1 as highest bias at stage 1. Now with g1 as input, let the highest bias output be g2.
Now overall bias would be 2*g1*g2

There might exists u1 bias(not greedy) at stage 1 (g1>u1), but with u1 as input we might get u2 a possible high bias output at stage 2 giving the product 2*u1*u2 > 2*g1*g2
So greedy bias at a stage may not give the highest overall bias because it restricts the section of inputs to stage2 which might have the potential to give high overall bias.

Time Complexity:
With a 3x3 Sbox, N=9 and T=3 the program completed in less than 0.5 seconds.

O((T+1)*2^N)+O(N/S * 2^N * 2^N * 2^S * S) with memoization, the second term is for calculating all the possible bias pairs in all rows.

O(2^(n*(T+1)) exponential complexity without memoization

At every stage, there would be 2^n * 2^n combinations possible at each sbox row.
For a total of T stages, There would be 2^(n*(T+1)) total possible paths from input to output.
Bias calculation

Sample Output:

```
3
9
0 3 6 1 4 7 2 5 8
3
0 2 4 6 3 1 7 5
P8, K08, K15, K18, K24, K25, K27, K28, C1, C2, C4, C5, C7, C8
Bias =  0.5
```