Question 2:

Algo:

For each input in stage, we find a feasible set of outputs from the precalculated input-output S-box mappings. After permutating this output (which will input for the next stage) we repeat this process.

In any stage, if we find that the bias till that stage from the top is less than the best bias then we abandon the current path. After reaching the last stage if we find that total bias is greater than best bias until now we update the best bias. This process is repeated for all inputs

Complexity

- Number of layers: $O(T)$
- Number of states in one layer of the dp: $O(2^N)$
  The input information is redundant and is not used in maximization.
- Preprocessing bias computation: $O(sx2^sx2^sx2^s) = O(sx2^{3S})$
  After that constant time access.
- Permutation, copy and other similar functions for each layer: $O(N)$
- Update each state of dp: $O(2^N)$
- Sbox preprocessing computation: $O(2^S)$

Time Complexity: $O(TxNx2^N + Nx2^{3s})$

“A greedy strategy will always work.” Examine the validity of this claim?

-> A greedy strategy **won’t always work.**

Counter Example:

T = 3
S = 4
N=8
S-Box = (0,1,2,3,5,6,7,4,10,11,8,9,12,13,14,15)
P-Box = (0,1,3,5,4,2,6,7)
input = 0b00100000

Using the greedy approach, we arrive at the path
P2 K02 K11 K13 K23 K33 C3 whose bias is 0.375 * 0.25 * 0.25 = 0.0234375

But the path with the highest bias in
P2 K02 K11 K12 K21 K31 C1 whose bias is 0.125*0.5*0.5 = 0.03125

Here we obtained the path with the highest bias when we chose a combination with lesser bias in the first round.