Online – 1

Greedy

Huffman coding.

Coreman section 16.3

Online – 2

Divide and Conquer

Local Minimal: The local minimal of a sequence

$$P_1, P_2, P_3, . . . , P_n$$

is P_i such that

1. $1 < i < n$
2. $P_{i-1} > P_i$ and $P_i < P_{i+1}$

Input:

1. An integer, n >= 3 representing the number of coins.
2. A sequence

   $$P_1, P_2, P_3, . . . , P_n$$

   Here, $P_1 > P_2$ & $P_n > P_{n-1}$

   NB: this condition ensures that there is always a local minimal.

Output:

A local minimal of the sequence.

Solution:

Similar to: https://drive.google.com/open?id=0By-BfovJ3XAWVXF6OUMwb0hyVG8

Online – 3

Dynamic Programming

Input:

3. An integer, n representing the number of coins.
4. A sequence

$$P_1, P_2, P_3, \ldots, P_n$$

   Here P_i = probability of getting head if i-th coin is flipped.
5. An integer, k <= n

Output:

1. The probability of getting exactly k heads if all the coins are flipped.
2. Print the DP table.

NB: the problem must be solved using bottom up approach.

Solution:

Recurrence relation: [see from offline folder]

T(n, k) = Probability if n th coin gives head + Probability if n th coin does not give head

$$= P_n * T(n - 1, k - 1) + (1 - P_n) * T(n - 1, k)$$

Base Cases:

1. `T(n, n)`
2. `T(n, 0)`

Code: http://ideone.com/ylt19F

Running Time: $O(nk)$

Online – 4

Branch & Bound

Input:

1. An integer, n, the number of processors
2. An array of times taken by some processes

$A_1, A_2, A_3, . . . , A_n$

Where Ai = time taken by i-th process

Output:

1. Output according to the greedy algorithm.

Solution:

[see greedy algorithm from offline – 4 folder]

Online – 5

Graph

Input:

1. An undirected unweighted graph
2. A node , u

Output:

1. Whether it has a spanning tree
2. Shortest path from the node u to all the other vertices.

Solution:

See the “online 5” folder.