118

COL351: Quiz-2 Name: VIRAJ AGASHE

Maximum marks: 5+5 = 10 Entry number: 2 0 2 0 C S 1 0 5 6 7

Question 1 Let $X, Y, Z \in \{0,1\}^n$ be three n-length strings. Describe an $O(n^2)$ time algorithm to compute largest k such that there exists a k-length string that is substring of X, Y, and Z.

~~Sol. We want to find LCS of X, Y, Z.~~

$x_1\ x_2 - - .\ x_n$
$y_1\ y_2 - -\ y_n$
$z_1\ z_2 - -\ z_n$

Sol. Algorithm:

Pass the string Y over the string X to find the set of all common substrings.

For each configuration, this takes $O(n-k)$ time, & n configurations possible.

$x_1\ x_2 - .\ x_n$

$y_1\ y_2 - -\ y_k - -\ y_n$

So overall it takes $O(n^2)$ time.

Further, for each of the substrings we get, we can treat it as a "pattern" & search for it in the string Z. $O(n^3)$.

By KMP algorithm, we know we can do this in $O(n+k)$ time, where k is the size of the pattern.

$\therefore$ we get an algorithm which is

$$O(n^2 + p\cdot(n+k))$$

where p is the no. of common strings.

Question 2 Let $G = (V, E)$ be a weighted digraph with no cycle of negative weight, and let $S \subseteq V$ be a set of size k. A path P is said to be an S-path if the internal vertices of P lie in S.

Describe an $O(kn^2)$ time algorithm to compute a binary matrix B such that $B[i, j] = 1$ if and only if there exists an S-path of non-negative weight from vertex i to vertex j in G.

Ans. Note that if there ∃ a S path of -ve weight, ~~it must also be -ve~~ shortest S-path must also be of -ve weight.

We modify Bellman-Ford as follows:

B[n][n] ← Initialize with 0

For each $v \in V$:

$D = (d_1, d_2 \ldots d_n)$

$D[k] = \infty \ \forall k$, $D[v] = 0$

For i in $(1 \ldots k)$:

~~if / for~~ for all $(v, w) \in S$:

if ($D[v] \Leftrightarrow D[w] + wt(v, w)$ and $(v, w) \in S$):

$D[v] = D[w] + wt(v, w)$

$O(nk)$

$O(n^2k)$

After k iterations we will know if ~~there is~~ shortest path exists using only ~~vertices~~ vertices in S

Now we can fill up the matrix B.

for all k s.t. ~~(D[k] < 0)~~ $(D[k] < 0)$:

$B[v][k] = 1$

$O(n)$

Finally, return B

The inner loop runs in $O(nk)$ time, and for each vertex we run it, so we get total time (once)

$= O(n^2k)$