# CS532 Homework 11
Archana Machireddy

# Question 1

T = 1583832647
P = 83832

Using Horner'r rule

p = P[m] + 10(P[m-1] +10(P[m-2] + … + 10(P[2] +10P[1])))))

p = 2 + 10(3 +10(8 +10(3 + 10*8)))
= 2 + 10(3 +10(8 +10(83)))
= 2 + 10(3 +10(838))
= 2 + 10(8383)
= 83832

$t_o$ = 8 + 10(3 +10(8 +10(5 + 10*1)))
= 8 + 10(3 +10(8 +10(15)))
= 8 + 10(3 +10(158))
= 8 + 10(1583)
= 15838

$t_{s+1} = 10(t_s – 10^{m-1}$ T[s+1]) + T[s+m+1]

$t_1 = 10(t_0 - 10^{5-1}$T[1]) + T[6]
= 10 (15838 – 10000*1) + 3
= 58380 + 3
= 58383
$t_2 = 10(t_1 - 10^{5-1}$T[2]) + T[7]
= 10 (58383 – 10000*5) + 2
= 83830 + 2
= 83832
$T_3 = 10(t_2 - 10^{5-1}$T[3]) + T[8]
= 10 (83832 – 10000*8) + 6
= 38320 + 6
= 38326

# Question 2

$2^{16}$ = 65536. Storing numbers from 0, largest number that can be stored is 65535.
The largest value of q is 6553.

If the chance of $t_s$ being equivalent to p mod q is 1/q, then the number of spurious hits can be O(n/q). As each time we have a spurious hit we check the entire pattern and a spurios hit can happen in O(n) positions, The expected match time for Rabin- Karp algorithm is O(n) +O(m(v+n/q)), where v is the number of times $t_s$ == p and all the characters match. If we choose a large value of q, then all our ts and p values will be within one computer word. So now it will take only O(1) time to check if these two values are equal. Which might not have been true if we were using the whole pattern to compare and m was large. So now the Rabin – Karp algorithm can run in O(n+m) and as m<=n, the matching time in O(n).

# Question 3

-25 mod 17
qc + r = x. Here the q is different from what we use in the algorithm.
c = 17, x = -25
when q = -1, r = -8
when q = -2, r = 9
as 9 < 17 , -25 mod 17 = 9.

T = 1583832647
P = 83832

New p and $t_o$ are:
p = p mod q = 83832 mod 6553 = 5196
$t_0$ = $t_0$ mod q = 15838 mod 6553 = 2732

$h = d^{m-1} \bmod q = 10^4 \bmod 6553 = 3447$

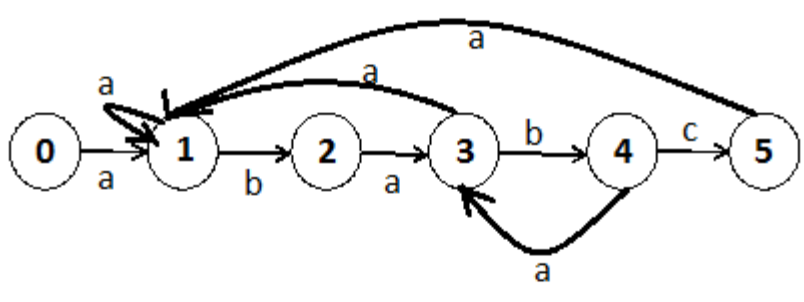
$t_{s+1} = (10(t_s – T[s+1]h) + T[s+m+1]) \bmod q$

$t_1 = 10(t_0 – T[1]h) + T[6] = (10\ (2732 – 1 * 3447) + 3\ ) \bmod 6553 = -7147 \bmod 6553 = 5959$

$t_2 = 10(t_1 – T[2]h) + T[7] = (10\ (5959 – 5 * 3447) + 2\ ) \bmod 6553 = (10\ (5959 – 17235) + 2\ ) \bmod 6553$
$= (10\ (-11276)\ ) + 2 \bmod 6553 = -112758 \bmod 6553 = 5196$

$t_3 = 10(t_2 – T[3]h) + T[8] = (10\ (5196 – 8 * 3447) + 6\ ) \bmod 6553 = -223794 \bmod 6553 = 5561$

# Question 4



# Question 5

| | Prefix |
|---|---|
| $P_0$ | ε |
| $P_1$ | a |
| $P_2$ | ab |
| $P_3$ | aba |
| $P_4$ | abab |
| $P_5$ | ababc |

# Question 6

| i | $P_i$ | Next Character | | | | | |
|---|---|---|---|---|---|---|---|
| | | a | | b | | c | |
| | | String | Longest Prefix | String | Longest Prefix | String | Longest Prefix |
| 0 | ε | εa | 1 | εb | 0 | εc | 0 |
| 1 | a | aa | 1 | ab | 2 | ac | 0 |
| 2 | ab | aba | 3 | abb | 0 | abc | 0 |
| 3 | aba | abaa | 1 | abab | 4 | abac | 0 |
| 4 | abab | ababa | 3 | ababb | 0 | ababc | 5 |
| P | ababc | ababca | 1 | ababcb | 0 | ababcc | 0 |

# Question 7

Shortest path from node 2 to 1 using just 1 edge is 8, and using 2 edges is 5 (2 -> 3 -> 1).

$$L^{(1)} = \begin{bmatrix} 0 & 2 & 5 & \infty \\ 8 & 0 & 7 & \infty \\ -2 & \infty & 0 & -4 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$l_{ij}^{(m)} = \min_{1 \le k \le n} \{ l_{ik}^{(m-1)} + w_{kj} \}$$

$$l_{21}^{(2)} = \min_{1 \le k \le 4} \{ l_{2k}^{(1)} + w_{k1} \}$$

$$l_{21}^{(m)} = \min\{(8+0), (0+8), (7-2), (\infty + \infty)\}$$
$$l_{21}^{(m)} = 5$$

# Question 8

$$L^{(2)} = \begin{bmatrix} 0 & 2 & 5 & 1 \\ 5 & 0 & 7 & 3 \\ -2 & -5 & 0 & -4 \\ 7 & -1 & 6 & 0 \end{bmatrix}$$

$$L^{(3)} = \begin{bmatrix} 0 & 0 & 5 & 1 \\ 5 & 0 & 7 & 3 \\ -2 & -5 & 0 & -4 \\ 4 & -1 & 6 & 0 \end{bmatrix}$$

# Question 9

$L^1 = D^0$

# Question 10

The shortest path from node 3 to node 2 in which all intermediate vertices are in {}, has no intermediate nodes, therefore the weight is ∞, as there is no direct path from 3 to 2. With intermediate vertices in {1}, it is 0, as there is path from 3 to 2 via 1. With intermediate vertices in {1,2,3,4}, the weight is -5, as you can reach 2 via vertex 4 now.

$$D^{(0)} = \begin{bmatrix} 0 & 2 & 5 & \infty \\ 8 & 0 & 7 & \infty \\ -2 & \infty & 0 & -4 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\} \quad for\ k \geq 1$$

$$d_{32}^{(1)} = \min\{d_{32}^{(0)}, d_{31}^{(0)} + d_{12}^{(0)}\}$$
$$d_{32}^{(1)} = \min\{\infty, (-2+2)\} = 0$$

The shortest path is the minimum of the weight of the path from node 3 to 2 without using the presently added vertex (1), or the weight from vertex 3 to 1, plus the weight from vertex 1 to 2.

# Question 11

$$D^{(1)} = \begin{bmatrix} 0 & 2 & 5 & \infty \\ 8 & 0 & 7 & \infty \\ -2 & 0 & 0 & -4 \\ \infty & -1 & \infty & 0 \end{bmatrix}$$

$$D^{(2)} = \begin{bmatrix} 0 & 2 & 5 & \infty \\ 8 & 0 & 7 & \infty \\ -2 & 0 & 0 & -4 \\ 7 & -1 & 6 & 0 \end{bmatrix}$$

$$D^{(3)} = \begin{bmatrix} 0 & 2 & 5 & 1 \\ 5 & 0 & 7 & 3 \\ -2 & 0 & 0 & -4 \\ 4 & -1 & 6 & 0 \end{bmatrix}$$

$$D^{(4)} = \begin{bmatrix} 0 & 0 & 5 & 1 \\ 5 & 0 & 7 & 3 \\ -2 & -5 & 0 & -4 \\ 4 & -1 & 6 & 0 \end{bmatrix}$$

# Question 12

Initialization:

| Node | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | $\infty$ | $\infty$ | $\infty$ | 0 |
| pi | Nil | Nil | Nil | Nil |

First iteration:
Going through the edges in the order (1,2) (1,3) (2,1) (2,3) (3,1) (3,4) (4,2)
Edge (4,2) relax is successful. Its weight is -1. Vertex 2 has been updated. Its new d is -1 and pi is vertex 4.

| Node | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | $\infty$ | $-1$ | $\infty$ | 0 |
| pi | Nil | 4 | Nil | Nil |

Second iteration:

Edge (2,1) relax is successful. Its weight is 8. Vertex 1 has been updated. Its new d is 7 and pi is vertex 2.
Edge (2,3) relax is successful. Its weight is 7. Vertex 3 has been updated. Its new d is 6 and pi is vertex 2
Edge (3,1) relax is successful. Its weight is -2. Vertex 1 has been updated. Its new d is 4 and pi is vertex 3.

| Node | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | 4 | $-1$ | 6 | 0 |
| pi | 3 | 4 | 2 | Nil |

Third iteration:

Edge (1,2) relax is successful. Its weight is 2. Vertex 2 has been updated. Its new d is 6 and pi is vertex 1.

| Node | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | 4 | $-1$ | 6 | 0 |
| pi | 3 | 4 | 2 | Nil |

# Question 13

Algorithm in section 24.2 in single-source shortest paths in directed acyclic graphs. But the graph given in the question has multiple cycles. This algorithm can only be applied to graphs with no cycles.

# Question 14

Initialization:

| Vertex | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | $\infty$ | $\infty$ | $\infty$ | 0 |
| pi | Nil | Nil | Nil | Nil |

Queue : { (4, 0), (1, ∞), (2, ∞), (3, ∞)}

Iteration 1:

Vertex 4 is extracted in line 5. Vetrex 2's priority is changed to -1.

| Vertex | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | $\infty$ | $-1$ | $\infty$ | 0 |
| pi | Nil | 4 | Nil | Nil |

Queue : { (2,-1), (1, ∞) (3, ∞)}

Iteration 2:

Vertex 2 is extracted in line 5. Vetrex 3's priority is changed to 6 and vertex 1's priority is changes to 7.

| Vertex | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | 7 | $-1$ | 6 | 0 |
| pi | 2 | 4 | 2 | Nil |

Queue : { (3,6) (1, 7)}

Iteration 3:

Vertex 3 is extracted in line 5. Vetrex 1's priority is changed to 4.

| Vertex | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | 4 | $-1$ | 6 | 0 |
| pi | 3 | 4 | 2 | Nil |

Queue : { (1, 4)}

Iteration 4:

Vertex 1 is extracted in line 5. No priorities get updated.

| Vertex | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| d | 4 | $-1$ | 6 | 0 |
| pi | 3 | 4 | 2 | Nil |

Queue : { }