CSC 3110 – Homework 5

1. Consider the problem of searching for genes in RNA sequences using the Boyer-Moore algorithm. An RNA sequence consists of a text on the alphabet {A,C,G,U} and the gene or gene segment is the pattern

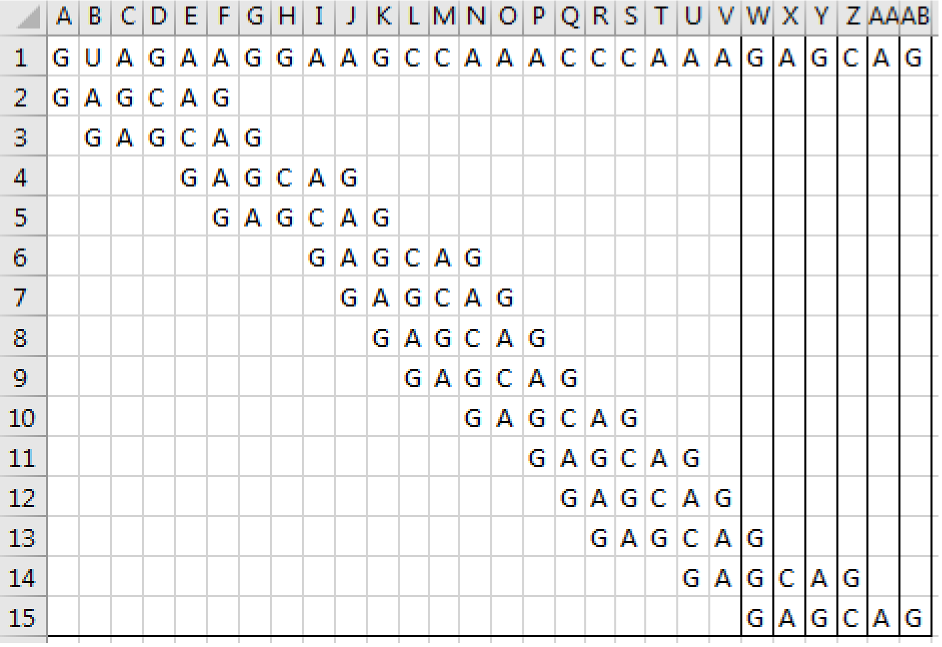
   a. Construct the bad symbol shift table for the pattern GAGCAG

G	A	C	*
3	1	2	6

   b. Construct the good suffix shift table for the pattern GAGCAG.

K	0	1	2	3	4	5	D2
1	G	A	G	C	A	G	5
2	G	A	G	C	A	G	3
3	G	A	G	C	A	G	3
4	G	A	G	C	A	G	3
5	G	A	G	C	A	G	3

   c. Apply the Boyer-Moore algorithm to locate GAGCAG in the following RNA sequence: GUAGAAGGAAGCCAAACCCAAAGAGCAG



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	G	U	A	G	A	A	G	G	A	A	G	C	C	A	A	A	C	C	C	A	A	A	G	A	G	C	A	G
2	G	A	G	C	A	G																						
3		G	A	G	C	A	G																					
4					G	A	G	C	A	G																		
5						G	A	G	C	A	G																	
6									G	A	G	C	A	G														
7										G	A	G	C	A	G													
8											G	A	G	C	A	G												
9												G	A	G	C	A	G											
10														G	A	G	C	A	G									
11																G	A	G	C	A	G							
12																	G	A	G	C	A	G						
13																		G	A	G	C	A	G					
14																					G	A	G	C	A	G		
15																							G	A	G	C	A	G

      i. When deciding how to shift the pattern, once a mismatch is found the value of the value of the mismatched character must be retrieved from the bad character table and the number of matches before it must be subtracted from it.(If that value is non-positive use 1 instead) If the mismatched character was not the first character the value associated with the length of the matching substring before it must be retrieved from the suffix table. The max of the two values determines how far to shift the pattern.

2. For the input 0, 49, 6, 13, 12, 3, 2, 1, 5, 4, 8, 21, 39 and hash function h(k)=k%13

a. Construct the open hash table

0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	2	3	4	5	6		8		49		12
13								21				
39												

b. Find the largest number of key comparisons in a successful search in this table

i. We can simply look at the table and see that the largest number of comparisons that would need to be made is 3 at the value of 0 in the table.

c. Find the average number of key comparisons in a successful search in this table

i. (1+α)/2 = (1+1)/2 = 1 (α is also known as the load factor which is # of keys / table size)

3. For the input 0, 49, 6, 13, 12, 3, 2, 1, 5, 4, 8, 21, 39 and hash function h(k)=k%13

a. Construct the closed hash table

0	1	2	3	4	5	6	7	8	9	10	11	12
0	13	2	3	1	5	6	4	8	21	49	39	12

b. Find the largest number of key comparisons in a successful search in this table

i. The largest number would be 12. If we were to search for 39 the hash function would return 0 but that cell is already occupied with another number so we'd have to go to the cell next to it and check. This process would repeat until we found an empty cell (unsuccessful) or the value (successful).

c. Find the average number of key comparisons in a successful search in this table

i. Summation of the number of comparisons for each key / number of keys = 2.4

1. 32/13 = 2.4

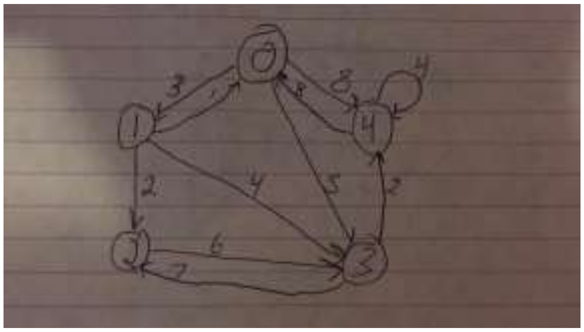
4. Apply the bottom-up dynamic programming algorithm to the following instance of the knapsack problem for a knapsack of capacity W=5

Item	Weight	Value
1	2	30
2	3	45
3	2	20

Weight	Value	Item	0	1	2	3	4	5
0	0	0	0	0	0	0	0	0
2	30	1	0	0	30	30	30	30
3	45	2	0	0	30	45	45	75
2	20	3	0	0	30	45	50	65

5. Solve the all-pairs shortest path problem for the digraph with the following weight matrix. Also draw the digraph whose weight matrix is given

0	3	∞	5	8
1	2	3	4	∞
∞	∞	0	6	∞
∞	∞	7	0	2
8	∞	∞	∞	4



0	3	∞	5	8
1	0	2	4	∞
∞	∞	0	6	∞
∞	∞	7	0	2
8	∞	∞	∞	4

0	3	∞	5	∞
1	0	2	4	9
∞	∞	0	6	∞
∞	∞	7	0	2
8	11	∞	14	4

0	3	5	5	12
1	0	2	4	9
∞	∞	0	6	∞
∞	∞	7	0	2
8	11	13	14	4

0	3	5	5	12
1	0	2	4	9
∞	∞	0	6	∞
∞	∞	7	0	2
8	11	13	14	4

0	3	5	5	7
1	0	2	4	6
16	19	0	6	8
10	13	7	0	2
8	11	13	14	4