Data Structures and Algorithms
INFO 6205
Homework 11
Due: April 14, 2019

Put all your java, compiled class files and documentation files into a zip file named Homework11.zip and submit it via the dropbox on the blackboard before the END of due date. Put your name on all .java files. There will be a short Quiz on this homework.

Lecture11Notes

Sequence Alignment with Dynamic Programming
https://web.stanford.edu/class/cs262/presentations/lecture2.pdf
http://pages.cs.wisc.edu/~bsettles/ibs08/lectures/02-alignment.pdf

Genetic Algorithm
http://www.cs.cmu.edu/~02317/slides/lec_8.pdf
https://annals-csis.org/Volume_5/pliks/249.pdf

Bellman-Ford Shortest-Path Algorithm
http://faculty.ycp.edu/~dbabcock/PastCourses/cs360/lectures/lecture21.html
Bellamn-Ford Shortest-path Negative Cycle
https://www.dyclassroom.com/graph/detecting-negative-cycle-using-bellman-ford-algorithm

---

1. Consider the sequences x = TACGGGTAT and y = GGACGTACG. Assume that the match score is +1, and the mismatch is -1, and gap penalties is -2.

A) Fill out the dynamic programming table for a global alignment between x and y.

		T	**A**	**C**	**G**	**G**	**G**	**T**	**A**	**T**
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
G	-2	-1	-3	-5	-5	-7	-9	-11	-13	-15
G	-4	-3	-2	-4	-4	-4	-6	-8	-10	-12
A	-6	-5	-2	-3	-5	-5	-5	-7	-7	-9
C	-8	-7	-4	-1	-3	-5	-6	-6	-8	-8
G	-10	-9	-6	-3	0	-2	-4	-6	-7	-9
T	-12	-9	-8	-5	-2	-1	-3	-3	-5	-6
A	-14	-11	-8	-7	-4	-3	-2	-4	-2	-4
C	-16	-13	-10	-7	-6	-5	-4	-3	-4	-3
G	-18	-15	-12	-9	-6	-5	-4	-5	-4	-5

B) Draw arrows in the cells to store traceback information.

		T	A	C	G	G	G	T	A	T
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
G	-2	-1	-3	-5	-5	-7	-9	-11	-13	-15
G	-4	-3	-2	-4	-4	-4	-6	-8	-10	-12
A	-6	-5	-2	-3	-5	-5	-5	-7	-7	-9
C	-8	-7	-4	-1	-3	-5	-6	-6	-8	-8
G	-10	-9	-6	-3	0	-2	-4	-6	-7	-9
T	-12	-9	-8	-5	-2	-1	-3	-3	-5	-6
A	-14	-11	-8	-7	-4	-3	-2	-4	-2	-4
C	-16	-13	-10	-7	-6	-5	-4	-3	-4	-3
G	-18	-15	-12	-9	-6	-5	-4	-5	-4	-5

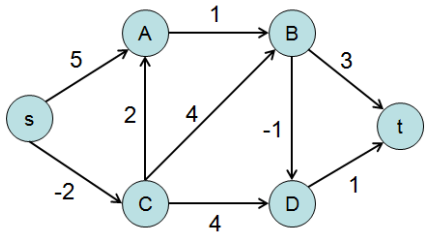
C) What is the score of the optimal global alignment and what alignment(s) achieves this score?

TACG
TACG

The score is -26 .

Note: Use references I sent you and covered them in class.

2. Consider the following Graph,



a) Solve the Shortest path of this graph using Bellman-Ford algorithm, step-by-step

s-a, s-c, a-b, b-t, b-d, c-a, c-b, c-d, d-t

	s	**A**	**B**	**C**	**D**	**t**
1st	0	INF	INF	INF	INF	INF
2st	0	0	2	-2	2	3
3st	0	0	1	-2	0	1

4st	0	0	1	-2	0	1
5st	0	0	1	-2	0	1

b) Write the Java code for the graph

c) Random Ordering is used to select the ordering of the algorithm steps.

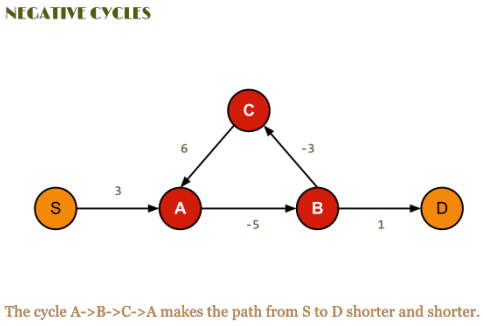
In the worse case, how many iterations are required by the algorithm to consider all Vertices?

It needs V – 1 times that get the result.

d) How to test if there is negative "cycle" in bellman-ford graph

If dist[v] > dist[u] + weight of edge uv, then "Graph contains negative weight cycle".

c) Consider the following bellman-ford graph, A negative "cycle" makes the path shorter and shorter: i) What does that mean? ii) How to test for negative cycle?



i) What does that mean?

It means there is a negative cycle, when we find the Shortest path, we cannot find the minimum path in the graph. Because it makes a negative cycle loop. That means make path shorter and shorter.

ii) How to test for negative cycle?

a. Initialize distances from source to all vertices as infinite and distance to source itself as 0. Create an array dist[] of size |V| with all values as infinite except dist[src] where src is source vertex.

b. This step calculates shortest distances. Do following |V|-1 times where |V| is the number of vertices in given graph.

   Do following for each edge u-v

   If dist[v] > dist[u] + weight of edge uv, then update dist[v]

   dist[v] = dist[u] + weight of edge uv

c. This step reports if there is a negative weight cycle in graph. Do following for each edge u-v

   If dist[v] > dist[u] + weight of edge uv, then "Graph contains negative weight cycle"

d. The idea of step 3 is, step 2 guarantees shortest distances if graph doesn't contain negative weight cycle. If we iterate through all edges one more time and get a shorter path for any vertex, then there is a negative weight cycle.

3. Read paper "Genetic Algorithms for Balanced Minimum Spanning Tree Problem".

   a) Read and understand only the first 5 pages.

      https://annals-csis.org/Volume_5/pliks/249.pdf

   b) Write Java code for the Algorithm described in paper on page-3.

   c) Compile and Run the code

   d) Describe the output results

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Contents
18 E-D-A-C-B
18 E-D-A-C-B
18 E-D-A-C-B
18 E-D-A-C-B
18 E-D-A-C-B
18 E-D-A-C-B
18 E-D-A-C-B
19 C-B-A-D-E
19 E-D-A-B-C
19 C-B-A-D-E
19 C-B-A-D-E
19 C-B-A-D-E
19 C-B-A-D-E
19 C-B-A-D-E
```

This result shows all the pathway to connect with each Point. The number means the weight of routes.