Solution Assignment 2

Maxime Descoteaux

February 19, 2004

Question 1 (10 pts)

We are given a sequence A = {a_1, a_2, ..., a_n}. We have to return the longest increasing and longest decreasing subsequence of A. We only need a 1-dimensional array, I or D, of length n to store the length of the longest subsequence. We also keep a predecessor array to return the actual subsequence (this was not required).
I[i] returns the length of the longuest increasing subsequece of {a_1, ..., a_i}.
D[i] returns the length of the longuest decreasing subsequece of {a_1, ..., a_i}.

```
LonguestIncreasingSubsequence(A) {
  for i = 1 to n
    I[i] = 1;
    p[i] = nil;

  for i = 2 to n
    for j = i-1 to 1

       if a_i >= a_j && I[j] + 1 > I[i]
          I[i] = I[j] + 1;
          p[i] = j;

  return I[n];
}

LonguestDecreasingSubsequence(A) {
  for i = 1 to n
    D[i] = 1;
    p[i] = nil;

  for i = 2 to n
    for j = i-1 to 1

       if a_i <= a_j && I[j] + 1 > I[i]
          I[i] = I[j] + 1;
          p[i] = j;

  return D[n];
}
```

The longest monotone subsequence is the maximum of the two outputted length. Use p array to backtrack and reconstruct the actual longest monotone subsequence.

Bonus (5 pts)

Given sequence $A = \{a_1, ..., a_n\}$. Consider elements a_i and a_{i-1} and compute I[i-1], D[i-1], I[i], and D[i], defined earlier. Note that for every i,

$$a_{i-1} \leq a_i \qquad \text{OR} \qquad a_{i-1} \geq a_i$$

which assures that

$$I[i] = I[i-1] + 1 \qquad \text{OR} \qquad D[i] = D[i-1] + 1$$

Hence, either I is incremented or D is incremented. Thus, the same pair I[i] and D[i] can *never* occur more than once (one of the two numbers would have to increase by one). So, there are $(k-1)^2$ possible different pairs of numbers I[i],D[i] such that each number is in the range $\{1, ..., k-1\}$. So, after at most $(k-1)^2 + 1$ terms in the original sequence A, a k must appear in I or D. This proves that every sequence of k^2 elements has at least a monotone subsequence of length k.

Question 2 (5 pts)

Direct proof:
If $\pi[s] \neq$ nil after some RELAX operations

$\Longrightarrow$ $d[u] + W(u,s) < d[s]$ for some vertex u

$\Longrightarrow$ $d[u] + W(u,s) < 0$ since the initialization step assigns d[s] a 0 value

$\Longrightarrow$ there exist a path from s back to s which has a total weight that is negative

$\Longrightarrow$ there is a negative cycle in G

Question 3 (10 pts)

The MAJORITY of you do not know how to run Bellman-Ford Algorithm!!! At EVERY iteration, you have to relax EVERY edge.

iteration	1	2	3	4	5
0	0	∞	∞	∞	∞
1	0	3	-1	2	-4
2	0	3	-3	2	-4
3	0	1	-3	2	-4
4	0	1	-3	2	-4

iteration 1

```
Relax
12 -> update -> d[2] = 3
13 -> update -> d[3] = 8
15 -> update -> d[5] = -4
24 -> update -> d[4] = 4
25 -> no update
32 -> no update
41 -> no update
43 -> update -> d[3] = -1
54 -> update -> d[4] = 2
```

iteration 2

```
Relax
12 -> no update
13 -> no update
15 -> no update
24 -> no update
25 -> no update
32 -> no update
41 -> no update
43 -> update -> d[3] = -3
54 -> no update
```

iteration 3

```
Relax
12 -> no update
13 -> no update
15 -> no update
24 -> no update
25 -> no update
32 -> update -> d[2] = 1
41 -> no update
43 -> no update
54 -> no update
```

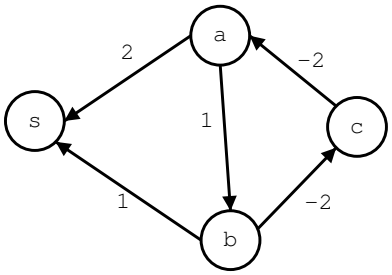

Figure 1: Conter example to professor's suggestion

iteration 4

no update

Question 4 (5 pts)

Consider the example in figure 1. If we use the professor's suggestion and run the algorithm with source s then we will never detect the negative weight cycle as the source does not reach it.

Prove that it works if G is strongly connected:

Suppose we have a strongly connected graph. Then, pick an arbitrary source is fine since from that source, we can reach every other vertex. Thus, if there exists a negative weight cycle in G, Bellman-Ford will detect it. Johnsson's output is correct.

If there is NO negative weight cycle then we proceed with the reweighing procedure. Since it does not depend on the weights of the graph and there is no negative weight cycle, all new $\hat{w}$ will be positive. Hence, the conditions for Dijsktra's algorithm to work are satisfied and Johnsson will return the correct answer.