MATH.APP.270 Algorithms for graphs

Programming assignment 2 **2022**

The Floyd Warshall algorithm given in the course notes has (at least) two deficiencies:

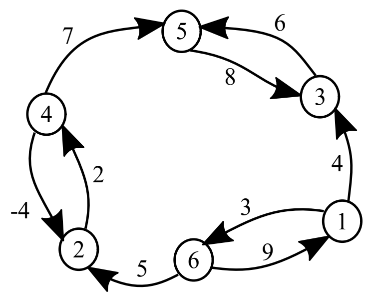
deficiency 1 It does not detect negatively weighted cycles and hence cannot make use of them when deciding if a minimal weighted paths from vertex u to vertex v exists.

deficiency 2 It does not produce minimal weighted paths from vertex u to vertex v, when such a path exists.

The purpose of this assignment is to make changes to the Floyd Warshall algorithm to correct these two deficiencies.

For our first example we will use the following digraph:

$G_1 =$



Using digraph G_1 as input to the Floyd Warshall algorithm of the course notes, we obtain the following D matrix:

$$D = \begin{bmatrix} 0 & 4 & 4 & 6 & 9 & 3 \\ \infty & -2 & 15 & 0 & 7 & \infty \\ \infty & \infty & 0 & \infty & 6 & \infty \\ \infty & -6 & 11 & -4 & 3 & \infty \\ \infty & \infty & 8 & \infty & 0 & \infty \\ 9 & 1 & 13 & 3 & 6 & 0 \end{bmatrix}$$

We will consider first how to use the results from the D to detect the existence of a minimal weighted path from vertex u to vertex v. This is then related to deficiency 1. We will use the notation $d[i][j]$ to indicate the element which is in the i'th row and the j'th column. Three conditions can be used:

condition 1 If $d[i][j] = \infty$, then there is no path from i to j and hence there is no minimal weighted path.

condition 2 If $d[i][i] < 0$, then vertex i belongs to a negatively weighted cycle. There are no minimal weighted paths starting from vertex i or ending with vertex i.

condition 3 Assume $d[i][j] \neq \infty$ and neither i nor j belong to a to a negatively weighted cycle. (Note: it is possible that $i = j$.) There does not exist a minimal weighted path from i to j, if there exists a vertex k having the following properties:

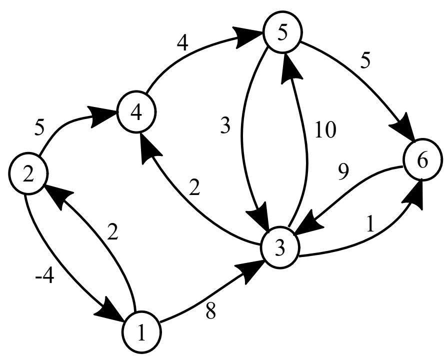
- k belongs to a negatively weighted cycle (hence $d[k][k] < 0$)
- there is a path from i to k (hence $d[i][k] \neq \infty$)
- there is a path from k to j (hence $d[k][j] \neq \infty$)

The following table contains the vertex pairs for which minimal weighted paths do not exist in digraph G_1 based on these three conditions:

condition	pairs (i, j) for which no minimal weighted path exists from i to j
1	$(2,1)$, $(2,6)$, $(3,1)$, $(3,2)$, $(3,4)$, $(3,6)$, $(4,1)$, $(4,6)$, $(5,1)$, $(5,2)$, $(5,4)$, $(5,6)$
2	$(1,2)$, $(1,4)$, $(2,3)$, $(2,4)$, $(2,5)$, $(4,2)$, $(4,3)$, $(4,5)$, $(6,2)$, $(6,4)$
3	$(1,3)$, $(1,5)$, $(6,3)$, $(6,5)$

For an example of condition 1 consider element $d[2][1]$. Since $d[2][1] = \infty$, there is no path from 2 to 1 and hence there is no minimal weighted path from 2 to 1. For an example of condition 2 consider element $d[1][2]$. Since $d[1][2] \neq \infty$ there is a path from 1 to 2. However since $d[2][2] < 0$, vertex 2 is part of a negatively weighted cycle. As such there is no minimal weighted path from 1 to 2. In fact, condition 2 tells us there is no minimal weighted path from 2 to any other vertex, nor from any other vertex to 2. For an example of condition 3 consider element $d[1][3]$. Since $d[1][3] \neq \infty$ there is a path from 1 to 3. Neither vertex 1 nor vertex 3 is part of a negatively weighted cycle, since $d[1][1] \geq 0$ and $d[3][3] \geq 0$. However, $d[1][2] \neq \infty$ and $d[2][3] \neq \infty$ and $d[2][2] < 0$. In other words there is a path from 1 to 3 that can go through a negatively weighted cycle and hence there is no minimal weighted path from 1 to 3. In fact, for digraph G_1 there are only 4 non-diagonal pairs for which there exist minimal weighted path: $(1,6)$, $(3,5)$, $(5,3)$ and $(6,1)$.

For our next example we consider the following digraph.

$G_2 =$ 

Using digraph G_2 as input to the Floyd Warshall algorithm of the course notes, we obtain the following D matrix:

$$D = \begin{bmatrix} -2 & 0 & 4 & 5 & 9 & 5 \\ -6 & -4 & 0 & 1 & 5 & 1 \\ \infty & \infty & 0 & 2 & 6 & 1 \\ \infty & \infty & 7 & 0 & 4 & 8 \\ \infty & \infty & 3 & 5 & 0 & 4 \\ \infty & \infty & 9 & 11 & 15 & 0 \end{bmatrix}$$

Based on the above 3 conditions it can be concluded there exist minimal weighted paths from u to v when u and v both belong to set $\{3,4,5,6\}$. For example, there is a minimal weighted path from 6 to 5 whose weight is $d[6][5] = 15$. However, owing to deficiency 2, the Floyd Warshall algorithm does not tell us what is the minimal weighted path. However, we can obtain this information if a parent matrix P is added to the Floyd Warshall algorithm. When computations are completed the $p[u][v]$ element from P has the following interpretation:

$$p[u][v] = \text{the predecessor of } v \text{ in the minimal weighted path from } u \text{ to } v\,. \qquad (1)$$

Near the end of the lecture on the Floyd Warshall algorithm a function PMINP is described which can used in producing the parent matrix P. The parent matrix for digraph G_2 is the following:

$$P = \begin{bmatrix} 2 & 1 & 1 & 2 & 4 & 3 \\ 2 & 1 & 1 & 2 & 4 & 3 \\ \infty & \infty & \infty & 3 & 4 & 3 \\ \infty & \infty & 5 & \infty & 4 & 3 \\ \infty & \infty & 5 & 3 & \infty & 3 \\ \infty & \infty & 6 & 3 & 4 & \infty \end{bmatrix}$$

Using this matrix we can now form minimal weighted paths, when such paths exist. For example, the minimal weighted path from 6 to 5 can be constructed using a **while**-loop:

```
x := 5
path := ⟨x⟩
while x ≠ 6 do
    x := p[6][x]
    add x to front of list path
end while
```

Using this patch of code we get the following minimal weighted path from 6 to 5:

$$\langle 6, 3, 4, 5 \rangle$$

Computing these minimal weighted paths, when they exist, we can obtain a *Path* matrix. Element $Path[i][j]$ is the minimal weighted path from i to j if such a path exists or the empty list if it does not exist.

For example, for digraph G_2 this *Path* matrix would contain the following:

$$Path = \begin{bmatrix} \langle\rangle & \langle\rangle & \langle\rangle & \langle\rangle & \langle\rangle & \langle\rangle \\ \langle\rangle & \langle\rangle & \langle\rangle & \langle\rangle & \langle\rangle & \langle\rangle \\ \langle\rangle & \langle\rangle & \langle 3\rangle & \langle 3,4\rangle & \langle 3,4,5\rangle & \langle 3,6\rangle \\ \langle\rangle & \langle\rangle & \langle 4,5,3\rangle & \langle 4\rangle & \langle 4,5\rangle & \langle 4,5,3,6\rangle \\ \langle\rangle & \langle\rangle & \langle 5,3\rangle & \langle 5,3,4\rangle & \langle 5\rangle & \langle 5,3,6\rangle \\ \langle\rangle & \langle\rangle & \langle 6,3\rangle & \langle 6,3,4\rangle & \langle 6,3,4,5\rangle & \langle 6\rangle \end{bmatrix}$$

Your task in this assignment is to write a method or function `allPathsFW`. This function should meet the following specifications:

- The input to `allPathsFW` is a weighted digraph.
- A modified Floyd Warshall algorithm should be used in `allPathsFW`.
- The end result of `allPathsFW` should be the *Path* matrix described above.

Data for testing
A set of graphs for testing purposes will be published separately.