Assignment 4.

Name: Guangfeng Lin
Student #: 301312131

Question 1:

algorithm:

Let G' = augmented G with s and its edges
if Bellman-Ford (G', s) is False
    return NULL;   // G has a negative-weight cycle.
else
    for each vertex v in V'
        $h(v) = \delta(s, v)$;

    for each edge (u, v) in E'
        $w^*(u,v) = w(u,v) + h(u) - h(v)$;

    for each vertex u in k disjoint subsets $V_1, V_2, \ldots, V_k$
        run Dijkstra (G, w^*, u);

        for each vertex set in G'
            $d_{uv} = \delta^*(u,v) + h(v) - h(u)$;

return distance;

Scanned with CamScanner

Question 2

a) An (s-t) flow is a function $f: E \to \mathbb{R}_{\geq 0}$ that satisfies the conservation constraint at every vertex v except possibly s and t:

$$\sum_u f(u \to v) = \sum_w f(v \to w).$$

Define $f(u \to v) = 0$ if there is no edge $u \to v$ in the graph. The value of the flow f, denoted $|f|$, the total net flow out of the source vertex s:

$$|f| := \sum_w f(s \to w) - \sum_u f(u \to s).$$

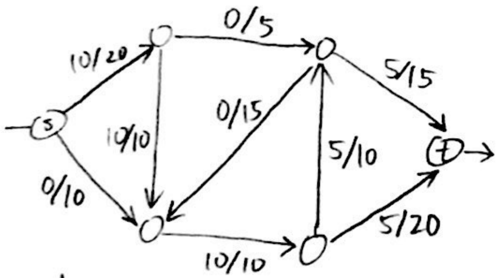
let $\partial f(v)$ denote the total net flow out of any vertex.

$$\partial f(v) = \partial f(s) + \partial f(t)$$
$$= \sum_u f(u \to v) - \sum_w f(v \to w)$$

Now, suppose we have another function $c: E \to \mathbb{R}_0 \geq 0$

f is feasible if $f(e) = c(e)$

Example:



Therefore, when all path from s to t are shortest-path distance, then the value of the maximum (s,t)-flows is at most E/d.

b) Graph $G = (V, E)$

for any vertices u and v either $c(u \to v) = 0$ or equivalently if an edge $u \to v$ replace the edge $u \to x \to v$ where x is a new vertex $c(v \to x)$ that $c(x \to v) = c(u \to v)$, and the modified graph has the same maximum flow value and minimum cut capacity as the original graph.

The Ford fulkerson algorithm runs $O(E|f^*|)$ or $O(VE^2)$ for worst time complexity.

Since the Edmonds-karp running time is $O(VE^2)$ for finding the total number of flow augmentations, so the maximum (s,t)-flow is at most $O(V^2/d^2)$.

Question 3

algorithm:

Input a similar bipartite graph that is derived from G as G'

G=(V, E) has a vertex set V and an edge set E.

Find the minimum cost perfect matching M for G'

$C = \{(u,v) : \{u, v'\} \in M, u \in V, v' \in V'\}$ // find the cycle

if C is empty

Return "No cycle cover exists".

else

Return C

Analysis:

The cycle covers in G is one-to-one correlation with perfect matches in G'

So the running time of this algorithm is $O(mn)$ which is a polynomial running time.

Question 4

$$\begin{pmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ -4 & 5 & 9 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 8 \\ -9 \end{pmatrix}$$

$$A = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ -4 & 5 & 9 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 8 \\ -9 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ -4 & 5 & 9 \end{bmatrix} = \begin{array}{c} \begin{matrix} 4 & -8 & 4 \end{matrix} \\ \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ -4 & 5 & 9 \end{bmatrix} \end{array} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ 0 & -3 & 13 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} 0 & 3 & -24 \\ 0 & -3 & 13 \end{matrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -\frac{3}{2} & 1 \end{bmatrix} \quad U = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Ly = Pb = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -\frac{3}{2} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 8 \\ -9 \end{bmatrix}.$$

$= y_1 = 0$

$y_2 = 8$

$y_3 = -9 + 0 + 12 = 3$

$$Ux = y = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -8 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 3 \end{bmatrix}.$$

$= x_1 - 2x_2 + x_3 = 0$

$2x_2 - 8x_3 = 8$

$\boxed{x_3 = 3}$

$2x_2 = 8 + 24$

$\boxed{x_2 = 16}$

$x_1 - 32 + 3 = 0$

$x_1 = 32 - 3$

$\boxed{x_1 = 29}$

$$x = \begin{pmatrix} 29 \\ 16 \\ 3 \end{pmatrix}$$