1.

Let edge $e \in G$ has change in capacity by 1 unit.

Step 1: Check if 'e' is utilized at full capacity i.e. $C_e - f_e = 0$ or not

Step 2: if $C_e - f_e \neq 0$, Output the same max flow as the original case. [as, $C_e \geq f_e$]

Step 3: if $C_e - f_e = 0$, it implies max-flow need to be updated. Use below subroutine to update max-flow.

3(i): pick one s-t path containing edge 'e' $\equiv (u, v)$ (say), such that each of the path's edges have atleast one-unit ongoing flow.

3(ii): Use Breath first search to find such s-t path.

3(iii): Remove one unit flow from each edge of the path.

3(iv): above step might create a new augmenting path. Use one iteration of Ford-Fulkerson algorithm to get max-flow. [Note: use updated capacity of edge 'e']

Step 4: output it as new max-flow.

Runtime Analysis: step-1 and 2 requires $O(1)$.

step 3(ii) requires BFS $\Rightarrow O(|E|+|V|)$

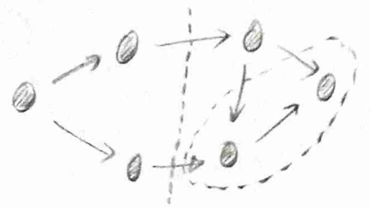
step 3(iv) requires running of one iteration of F-F algorithm. It demands DFS in network, once. $\Rightarrow O(|E|+|V|)$

Correctness argument: Requirement of step 3(iii) is essential to make up one unit extra flow passing in the new network along this particular s-t path.

Step 3(iv) requires a single iteration of F-F algorithm as just a single unit of flow has been decreased from any edge. No more augmenting path after one iteration.

2. Let the capacity function of original network 'G' be-

$$C : e \in G \rightarrow \mathbb{Z}^+$$



A transformation in capacity function C, such that min-flow in modified graph (G') corresponds to min-flow with least edge set.

Transformation :

$$C' : e' \in G \rightarrow \mathbb{Z}^+$$

$$C' \equiv |E|\, C + 1 \quad ; \quad |E| = \text{Total number of edges.}$$

Correctness argument :

Value of min-cut in c' follows below relationship

$$\text{min-cut}(c') = \sum_{P} (|E|\, C + 1) = |E| \sum_{P} C + \sum_{P} 1 \rightarrow (1)$$

[where 'p' is set of edges in the cut]

$$= |E|\, \text{min cut}(C) + |P|$$

Things to note: (1) $|E| > |P|$ for any min-cut edge set 'P'.

(2) minimization of min-cut (C') implies minimization of right side of equation (1)

$|E| \sum_{P} C \rightarrow$ minimizes if $\sum_{P} C$ is minimum, hence 'P' must be a mincut in original G.

$\sum_{P} 1 \rightarrow$ if it contains least number of edges.

3. a)

In this case, we need to find the path having maximum valued bottle neck path.

Hence, we keep track of bottleneck edge while traversing the edges in the Dijkastra.

Bottleneck edge → minimum capacity edge.

- Initialization :

  → each vertex given lowest possible value
  $c[V] = -\infty \; ; \; V \in G$

- updation of priority Queue :

  take out vertex from it that has max value
  (This ensure maximum capacity edge is choosen)

- Neighbour updation :

  go through all neighbour of v of ~~the~~ vetex (say, u)

  if $c[v] < \min(c[u], edge(u,v))$

  then,
  $c[v]$ updated to $\min\left(c[u], edge(u,v)\right)$

  [This ensure minimum capacity edge among the s-t path is traced].

Rest of the structure of the algorithm remain same.

(3b.)

→ There must be an augementing path since max-flow is still to be attained.

→ maximum no of edges in any augementing path $\leq |E| = m$

→ minimum no. of edges in any augementing path ≥ 1

$\boxed{\text{maximum no. of edges in min-cut} \leq |E|}$

since,

$$\sum_{P \to \text{min-cut}} c(e) = \text{max-flow}$$

→ flow 'F' has passed

$(F - F^*)$ has to be passed through

maximum 'm' edges.

→ Hence, each edges must have

atleast $\dfrac{(F-F^*)}{m}$ edge capacity.

It also implies a lower bound for increase in flow after each iteration.

3C.

Initially, assume there is no flow in the system.

so, maximum F^* can be pushed.

from problem 3.(b), we know atleast $(\frac{1}{m})$th part of this could be pushed regardless of augmenting path choosen

after one iteration,

new updated flow that can be pushed

$$= F^* - \frac{F^*}{m} = \left(1 - \frac{1}{m}\right) F^*$$

let after 'i' iterations, remaining flow to be pushed.

$$= \left(1 - \frac{1}{m}\right)^i F^*$$

Since, this is a product of two positive quantity. It can't be equal. We want it be less than 1. Once it become lesser than zero, we get an indication than the process is at its termination.

$$\left(1 - \frac{1}{m}\right)^i F^* < 1$$

$$i \cdot \log\left(1 - \frac{1}{m}\right) + \log F^* < 0$$

using, $\log(1-x) = -x - \frac{x^2}{2} \cdots$ [Tylor]

$$\Rightarrow \log\left(1 - \frac{1}{m}\right) < -\frac{1}{m}$$

$i > m \log F^* \Rightarrow (m \log F^*)$ iteration is sufficient

$$\boxed{\text{asymptotic scaling} : O(m \log F^*)}$$

3 d.

From problem 3.c, we get asymptotic scaling of the problem as $O(m \cdot \log F^*)$

max-flow $F^* < U \cdot |V| = U \cdot n$

[any min-cut must be lesser than total number of edges in the graph]

Hence,

$$O(m \cdot \log(U \cdot n)) \sim O(m \log U + m \log n)$$

$\log U \rightarrow$ number of bits to represent value U.

$\Rightarrow$ Runtime is polynomial wrt input size.