1. What's a subnet? How does a network get IP address for itself (network part of address)?

2. What is Border Gateway Protocol (**BGP**)? What are the guidelines for its selection of routes?

3. Assume that the routing table of Router B in the network has the following items (these three columns are represented as "destination network", "distance", and the next router, respectively):

N_1	7	A
N_2	2	C
N_6	8	F
N_8	4	E
N_9	4	F

Now B receives the routing information from C (these two columns are represented as "destination network" and "distance", respectively):

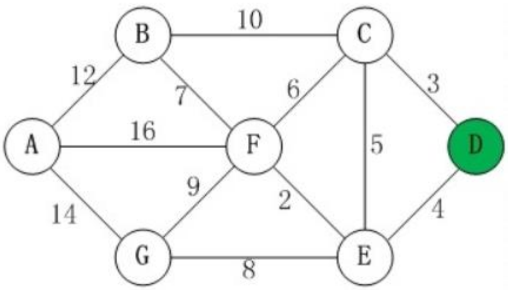
N_2	4
N_3	8
N_6	4
N_8	3
N_9	5

Please find the updated routing table of Router B according to the **Distance vector algorithm:**

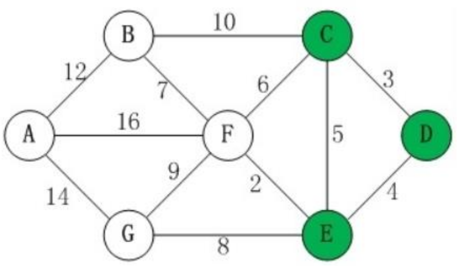
$$D_x(y) = \min_v\{c_{x,v} + D_v(y)\}$$

(Please show the detailed description of each step).

4. **DIJKSTRA'S LINK-STATE ROUTING ALGORITHM:**
Consider the graph on the left and the use of Dijkstra's algorithm to compute a least cost path from **D** to all destinations. Suppose that nodes **C** and **E** have already been added to **N'** (Figure2).



(Figure1)



(Figure2)

What is the next node to be added to **N'**? (F, B or G?) What is the shortest distance from D to A? (Please use the Dijestra algorithm according to the problem-solving steps taught by the teacher in class.)

*******Auxiliary information *******

Distance vector algorithm

Based on *Bellman-Ford* (BF) equation (dynamic programming):

Bellman-Ford equation

Let $D_x(y)$: cost of least-cost path from x to y.
Then:

$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

v's estimated least-cost-path cost to y

min taken over all neighbors v of x

direct cost of link from x to v

Network Layer: 5-18

Dijkstra's link-state routing algorithm

```
Initialization:
  N' = {u}                          /* compute least cost path from u to all other nodes */
  for all nodes v
    if v adjacent to u              /* u initially knows direct-path-cost only to direct neighbors */
       then D(v) = c_{u,v}          /* but may not be minimum cost! */
    else D(v) = ∞

Loop
   find w not in N' such that D(w) is a minimum
   add w to N'
   update D(v) for all v adjacent to w and not in N' :
       D(v) = min ( D(v), D(w) + c_{w,v} )
   /* new least-path-cost to v is either old least-cost-path to v or known
   least-cost-path to w plus direct-cost from w to v */
until all nodes in N'
```

Network Layer: 5-12