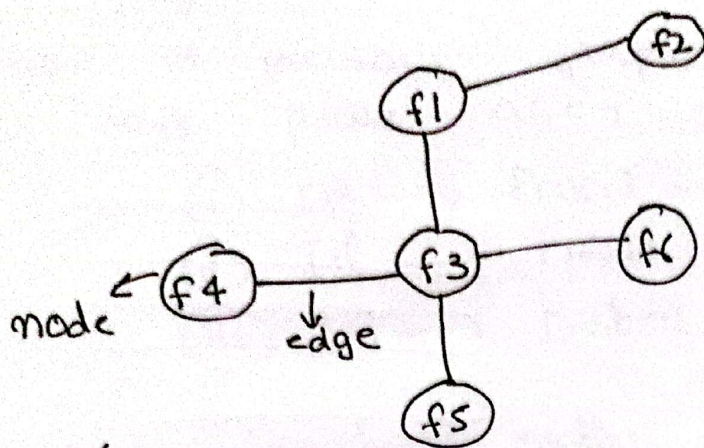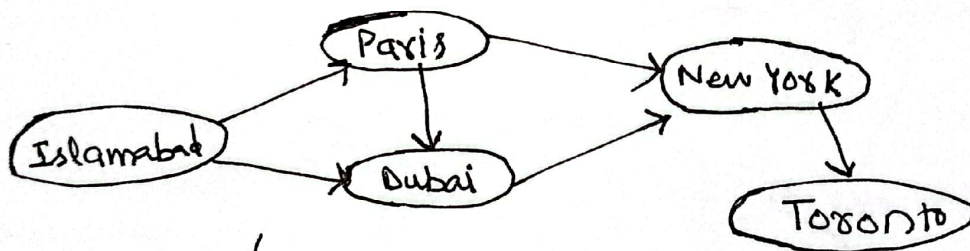(12) **Graph :**

→ Here is a facebook
   connection
   where f3 have
   friends f1, f4,
   f5, f6.

→ And f1 has
   a friend f2.

↳ This is <mark>undirected</mark> graph means there
   is no particular direction.

Another example :

↳ This is a <mark>directed</mark> graph
   as it shows directions.

<u>How</u> <mark>Graph</mark> and <mark>Tree</mark> are <mark>different</mark> :

↳ In Trees you will have only one
   connections between nodes. But
   on the hand in graph as you
   see above to reach newYork
   you can go Paris ⟶ New York
        or Dubai ⟶ New York
        or Paris ⟶ Dubai ⟶ New York.

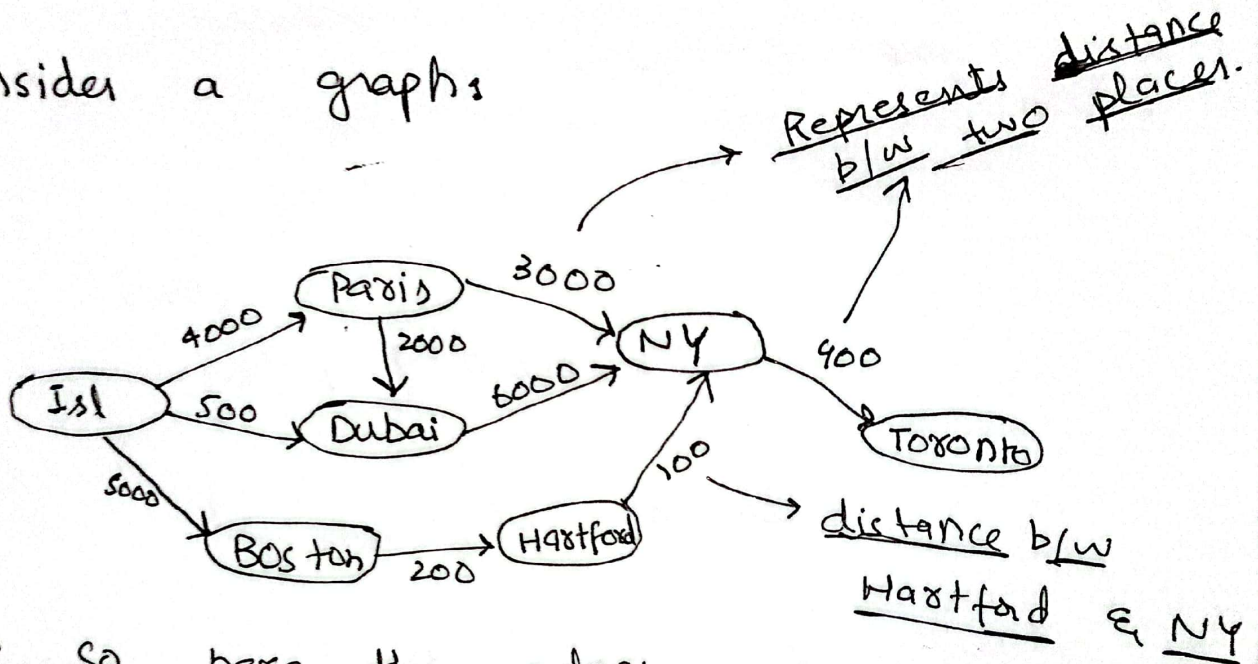→ Graphs are used to find paths b/w routes – Also we can find shortest path.

Like in previous graph to reach new York paths are:

① Isl ⟶ Paris ⟶ New York
② Isl ⟶ Paris ⟶ Dubai ⟶ NY
③ Isl ⟶ Dubai ⟶ NY

↳ And shortest paths are:

① Isl ⟶ Paris ⟶ NY
② Isl ⟶ Dubai ⟶ NY

Consider a graph:

So here the edges are weighted on base of distance so this is called as **Weighted Graph.**

So NOW Shortest Path to reach NY (on base of distance)

Isl $\xrightarrow{5000}$ Boston $\xrightarrow{200}$ Hartford $\xrightarrow{100}$ NY

↳ Total = 5300

→ Google Maps uses graphs behind
→   Facebook

Implementation Tips

⤷ we can represent connected nodes like
   Isl as Dubai as a Tupple
   as it is acting like pair.
        like   (Isl, Dubai)

⤷ Processing these tupple will be difficult
   for methods of find routes
   so what we can do is
        ⤷ Transform Tupple in the way
        that each node is key and
        it has value (list) of nodes
            connected with it.

   like:
        'Isl' : ['Dubai', 'Paris']
            ⋮
            ⤷ similarly for all

⤷ **How we will Transform Tuples
    to Dictionaries :**

⤷ If we are getting (Isl, Dubai) →
    we will make Isl as key & Dubai
    as value.
    ⤷ After that we get (Isl, Paris) →
    we will simply append Paris
    with Dubai.