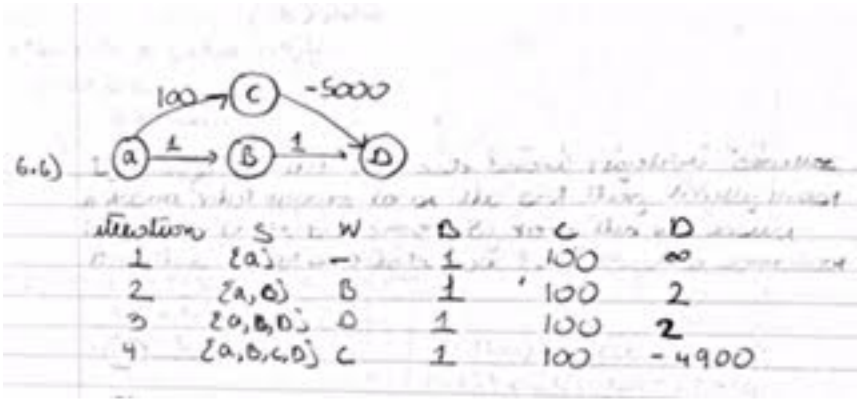
Problem 6.6



iteration	S	W	B	C	D
1	{a}	–	1	100	∞
2	{a, b}	B	1	100	2
3	{a, B, D}	D	1	100	2
4	{a, B, C, D}	C	1	100	-4900

In the above case and in accordance to the constraints placed by dijkstra's algorithm, at each iteration, an assumption is made that all previous iterations are correct and when we add a node to the set and set the distance to dist[v], we must infer this distance to be optimal. If this distance is not optimal, there must be some other path to the vertex V that is of shorter length.

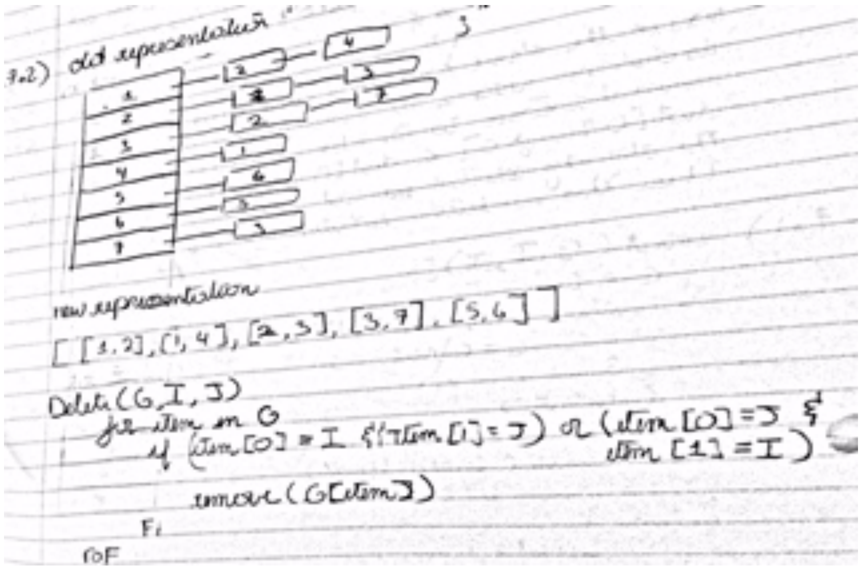
Now in the above case, when we added D to the set after its distance computed was 2, we assumed that the distance was the most minimal distance to D. When we add node C to the set, dist[D through C] < dist[D through A and b] even though the already shortest cost to D was already determined. Thus, this condition violates Dijkstra's arguments.

Problem 7.1

```
Insert(G, I, J):
   C = G[I];
      While(C.next != null)
         C = C.next
  C.next = J
  C = G[J]
     While(C.next != null)
         C = C.next
  C.next = I

Delete(adjlist, I, J)
   For key in adjlist
      If key is I or key is J
         Set.add(getindex(key))

  Ptr1 = adjlist(set[0])
  Ptr2 = adjlist(set[1])

  While(1):
     If Ptr1.next is I or J
```

```
        If Ptr1.next.next == null
            Ptr1.next == null
        Else
            Ptr1.next = Ptr1.next.next
    If Ptr1 != null
       Ptr1 = Ptr1.next
    If Ptr2.next is I or J
        If Ptr2.next.next == null
            Ptr2.next == null
        Else
            Ptr2.next = Ptr2.next.next
    If Ptr2!= null
       Ptr2= Ptr1.next

   If Ptr1 and Ptr2 are null
      Break out of while loop
```

7.2



1
2
3
4
5
6
7
2
4
3
3
7
2
1
6
3
3

7.2) old representation

new representation

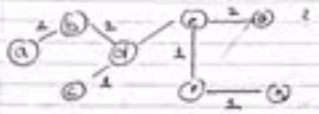
[[1,2], [1,4], [2,3], [3,7], [5,6]]

```
Delete(G, I, J)
    for item in G
        if (item[0] = I & item[1] = J) or (item[0] = J &
                                            item[1] = I)
            remove(G[item])
        Fi
    roF
```

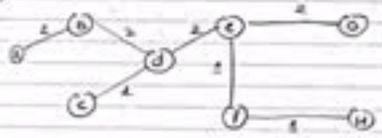
7.3

7.3)

a) Prim



b)



weight	source	dest	
1	e	d	–
1	e	F	–
1	F	H	–
2	a	b	–
2	b	d	–
2	d	c	–
2	e	g	–
2	F	h	–
3	a	c	–
3	h	g	–
4	d	F	

D) Breadth first search