Assignment#5 - Solution

Question 1:

a) DFS
Order of vertex visit: 1,2,3,4,5,6,8,7
Discovery edges in order of labeling: (1,2) (2,3)(3,4)(4,5)(5,6)(6,8)(5,7)
Back edges in order of labeling: (3,1)(5,3)(6,3)(8,4)(7,4)

b) BFS
Order of vertex visit: 1,2,3,4,5,6,7,8
Lists Li: L0={1}, L1={2,3}, L3={4,5,6}, L4={7,8}:
Discovery edges in order of labeling: (1,2) (1,3)(3,4)(3,5)(3,6)(4,7)(4,8)
Cross edges in order of labeling: (2,3) (4,5) (5,6)(5,7)(6,8)

Question 2: Dijkstra

a)

New Vertex	A	B	C	D	E	F	New Edge
A	0	5	∞	10	∞	4	---
F	0	5	∞	9	∞	4	(A,F)
B	0	5	7	8	∞	4	(A,B)
C	0	5	7	8	9	4	(B,C)
D	0	5	7	8	9	4	(B,D)
E	0	5	7	8	9	4	(C,E)

b) weight of shortest path spanning tree = 16

Question 3:

a) Prim-Jarnik algorithm:
Edges in order they are added to MST (direction important):
(A,F) (F,D) (D,E)(E,C)(C,B)
or (A,F) (A,B) (B,C)(C,E)(E,D)

Total weight of the MST=14

b) Kruskal's algorithm
Edges in order they are added to MST:
(D,E),(B,C),(C,E),(A,F),(A,B)
or (D,E),(C,E),(B,C),(A,F),(A,B)
or (D,E),(B,C),(C,E),(A,F),(D,F)
or (D,E),(C,E),(B,C),(A,F),(D,F)
Total weight of the MST=14

Question 4:
a) chaining:

0	1	2	3	4	5	6	7	8	9	10
		13	3	15	5					
			>25	>4	>16					

Number of probes to find each key: 5:1, 3:1, 15:1,25:2, 4:2, 13:1, 16:2

Average number of probes to find a key: $10/7 \sim 1.428\ldots$

b) quadratic probing:

0	1	2	3	4	5	6	7	8	9	10
		13	3	15	5	16	25	4		

Number of probes to find each key: 5:1, 3:1, 15:1,25:3, 4:3, 13:1, 16:2

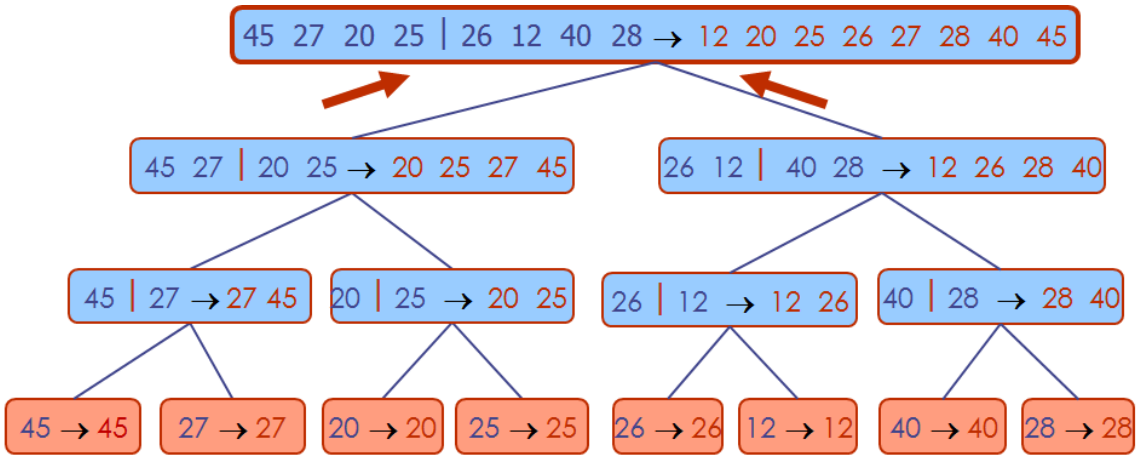
Average number of probes to find a key: $12/7 \sim 1.71\ldots$

c) double hashing:

0	1	2	3	4	5	6	7	8	9	10
16		13	3	15	5			25	4	

Number of probes to find each key: 5:1, 3:1, 15:1,25:2, 4:2, 13:1, 16:3
Average number of probes to find a key: $11/7 \sim 1.57\ldots$

a) Mergesort tree: (feel free to continue tree in text mode)



b) Quicksort table: (feel free to add more rows)

a	b	S
-	-	[45, 27, 20, 25, 26, 12, 40, 28]
0	7	12 27 20 25 26 28 40 45
0	4	12 25 20 26 27 28 40 45
0	2	12 20 25 26 27 28 40 45
6	7	12 20 25 26 27 28 40 45

NB: The trace is confirmed by the output of the following code.

```
public class Question5c  {

   public static void main(String[] args) {

      int[] s={12, 27, 20, 25, 26, 45, 40, 28};

      quicksort(s,0,7);


   }
   public static void quicksort (int [] s, int a, int b){
      int left=a;
      int right=b-1;
      int pivot=s[b];
```

```
        int temp;
        if (a>=b) return;
        while (left <= right){
           while ((left <= right) && (s[left] < pivot)) left++;
           while ((left <= right) && (s[right] > pivot)) right--;
           if (left <= right){
              temp=s[left];
              s[left]=s[right];
              s[right]=temp;
              left++;
              right--;

           }
        }
        temp=s[left];
        s[left]=s[b];
        s[b]=temp;
        System.out.print("a:");
        System.out.print(a);
        System.out.print("  ");
        System.out.print("b:");
        System.out.print(b);
        System.out.print("  ");
        System.out.print("s:");

        System.out.print("  ");
        System.out.print(s[0]);
        System.out.print("  ");
        System.out.print(s[1]);
        System.out.print("  ");
        System.out.print(s[2]);
        System.out.print("  ");
        System.out.print(s[3]);
        System.out.print("  ");
        System.out.print(s[4]);
        System.out.print("  ");
        System.out.print(s[5]);
        System.out.print(" ");
        System.out.print(s[6]);
        System.out.print(" ");
        System.out.println(s[7]);
        quicksort(s,a,left-1);
        quicksort(s,left+1,b);

    }
}
```