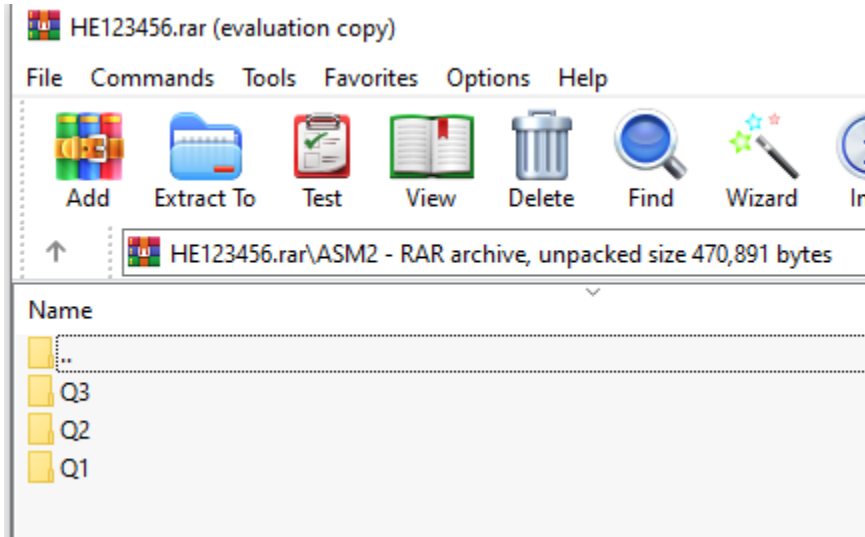


Lưu ý khi nộp bài:

Nén cả thư mục ASM2, sau đó đổi tên file nén thành mã sinh viên như hình dưới.

- KHÔNG được đổi tên các thư mục Q1, Q2, Q3
- KHÔNG được xóa, thay đổi nội dung file data.txt trong thư mục src. Có thể thay đổi nội dung của file data.txt ở thư mục ngoài để test chương trình.



Bài nộp mà vi phạm các yêu cầu trên sẽ không được chấm (nhận điểm 0).

Question 1: (4 marks)

(Do not pay attention to the real meaning of objects, variables and their values in the questions below).

In this question you should complete some methods in **MyList.java** file.

The class Car with 2 data members: owner and price is given and you do not need to edit it. The MyList class is a linked list of Car objects. The following methods should be completed:

- void addHead(String xOwner, int xPrice) - check if the last character of xOwner = 'B' or xPrice>100 then **do nothing**, otherwise add new car with owner=xOwner, price=xPrice to the head of the list. (price can get arbitrary value, even negative).

- void **f1()** – This method is used to test the addHead method above. You do not need to edit this function.

(AH,10) (AC,4) (AF,4) (AE,6) (AD,2) (AC,7) (AA,9)

- void **f2()** – There is a given object x. You should write statements so that x will be inserted before the first element which has a price greater than 10.

(AH,1) (AC,4) (AF,4) (AE,6) (AD,12) (AC,7) (AA,9)

(AH,1) (AC,4) (AF,4) (AE,6) (X,1) (**AD,12**) (AC,7) (AA,9)

- void **f3()** – There is a given object x. You should write statements so that x will be inserted before the second element which has price is greater than 10.

Content of the file f1.txt:

(AE,19) (AC,2) (AF,4) (AE,6) (AD,12) (AC,7) (AA,9)

(AE,19) (AC,2) (AF,4) (AE,6) (X,1) (**AD,12**) (AC,7) (AA,9)

- void **f4()** – There is a given object x. You should write statements so that x will be inserted before the last element which has price is greater than 10.

(AE,19) (AC,2) (AF,4) (EE,6) (AD,1) (BC,8) (AA,9)

(X,1) (**AE,19**) (AC,2) (AF,4) (EE,6) (AD,1) (BC,8) (AA,9)

- void **f5()** – Suppose the list contains at least 3 elements. Delete the Node **after the first** node having price>10 and the second character of owner equal 'C'.

(AE,19) (AC,2) (AF,4) (EC,1) (AD,1) (BC,18) (**AC,19**)

(AE,19) (AC,2) (AF,4) (EC,1) (AD,1) (BC,18)

- void **f6()** – Suppose the list contains at least 3 elements. Delete the Node after the last node having price>10 and the second character of the owner equals 'C'.

(AE,19) (AC,2) (AF,4) (EC,11) (AD,1) (BC,18) (AC,9)

(AE,19) (AC,2) (AF,4) (EC,11) (AD,1) (BC,18)

- void **f7()** – Suppose the list contains at least 3 elements. Delete the Node **before** the first node having price>10 and the second character of owner equal 'C'.

(AE,19) (AC,2) (AF,4) (EC,11) (AD,1) (BC,18) (AC,9)

(AE,19) (AC,2) (EC,11) (AD,1) (BC,18) (AC,9)

- void **f8()** – Suppose the list contains at least 3 elements. Delete the Node **before** the last node having price>10.

(**AE,9**) (AC,20) (AF,4) (EC,1) (AD,1) (BC,8) (AC,9)

(AC,20) (AF,4) (EC,1) (AD,1) (BC,8) (AC,9)

- void **f9()** – Sort ascendingly by owner from the head to the last max value of price of the list. And sort descending by price of remaining elements.

(AE,9) (AC,2) (AF,4) (EC,1) (**AD,18**) (AH,1) (AA,3) (BC,8) (AC,9)

(EC,1) (AC,2) (AF,4) (AE,9) (AD,18) (AC,9) (BC,8) (AA,3) (AH,1)

- void **f10()** – Sort descending by owner from the head to the first minimum value of price of the list. And sort ascending by owner of remaining elements.

(AE,9) (AC,2) (AF,4) (EC,1) (AD,18) (AH,1) (AA,3) (BC,8) (AC,9)

(EC,1) (AF,4) (AE,9) (AC,2) (AA,3) (AC,9) (AD,18) (AH,1) (BC,8)

Question 2: (4 marks)

In this question you should complete some methods in **BSTree.java** files.

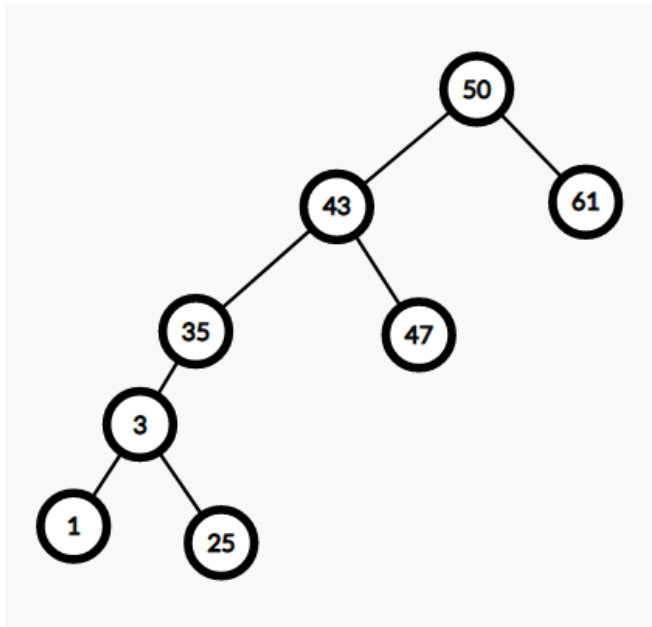
The class Car with 2 data members: owner and price is given and you do not need to edit it. The BSTree class is a binary search tree of Car objects. The variable **price is the key of the tree**. The following methods should be completed:

- void insert(string xOwner, int xPrice) - check if the second character of xOwner equals 'A' or xPrice >100 then **do nothing**, otherwise insert new car with owner=xOwner, price=xPrice to the tree.
- void **f1()** – You do not need to edit this function. Your task is to complete the insert(...) function above only.

Content of the file f1.txt:

(AB,50) (EB,43) (BD,61) (G1,35) (X2,47) (A2,3) (A1,1) (B2,25)

(AB,50) (EB,43) (BD,61) (G1,35) (X2,47) (A2,3) (A1,1) (B2,25)



- void **f2()** – Perform in-order traversal from the root but display the nodes having price in the interval [3,50] only.

Content of the file f1.txt:

(A1,1) (A2,3) (B2,25) (G1,35) (EB,43) (X2,47) (AB,50) (BD,61)

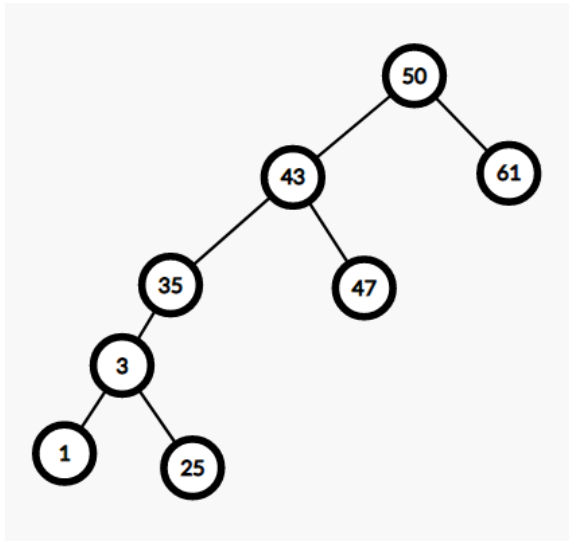
(A2,3) (B2,25) (G1,35) (EB,43) (X2,47) (AB,50)

- void **f3()** – Perform postOrder traversal from the root and delete by copying the first node having both 2 sons and price in the interval [30,70] .

Content of the file f1.txt:

(A1,1) (B2,25) (A2,3) (G1,35) (X2,47) (**EB,43**) (BD,61) (AB,50)

(A1,1) (B2,25) (A2,3) (X2,47) (G1,35) (BD,61) (AB,50)

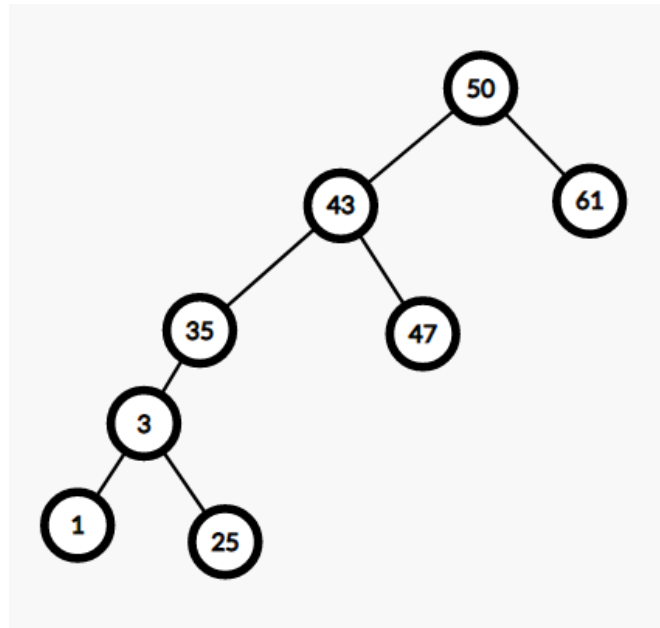


- void **f4()** – Perform inOrder traversal from the root and delete by Merging the first node having left son and price in the interval [30,70].

Content of the file f1.txt:

(A1,1) (A2,3) (B2,25) (G1,35) (EB,43) (X2,47) (AB,50) (BD,61)

(A1,1) (A2,3) (B2,25) (EB,43) (X2,47) (AB,50) (BD,61)

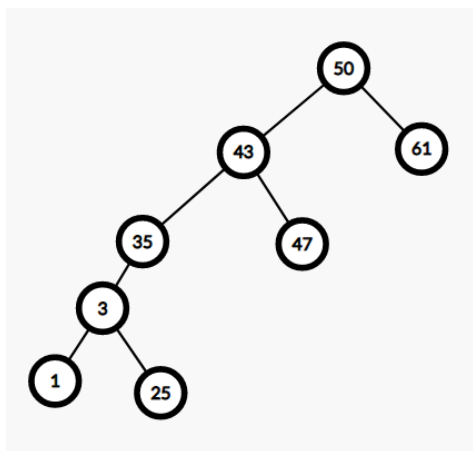


- void **f5()** – Suppose p is 4-th node when perform pre_Order traversal from the root of the tree and delete by Merging node p.

Content of the file f1.txt:

(AB,50) (EB,43) (G1,35) (A2,3) (A1,1) (B2,25) (X2,47) (BD,61)

(AB,50) (EB,43) (G1,35) (A1,1) (B2,25) (X2,47) (BD,61)

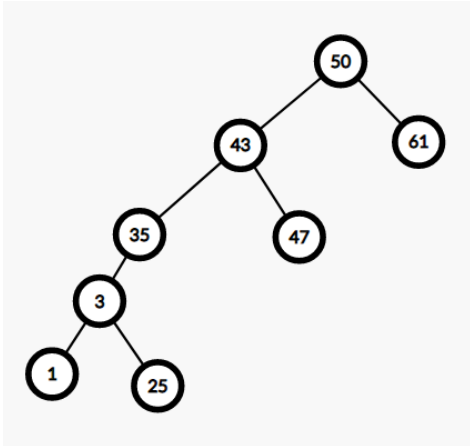


- void **f6()** – Suppose p is the 2-th node which has left node when performing Bread_first traversal from the root of the tree then delete by Merging node p.

Content of the file f1.txt:

(AB,50) (**EB,43**) (BD,61) (G1,35) (X2,47) (A2,3) (A1,1) (B2,25)

(AB,50) (G1,35) (BD,61) (A2,3) (X2,47) (A1,1) (B2,25)

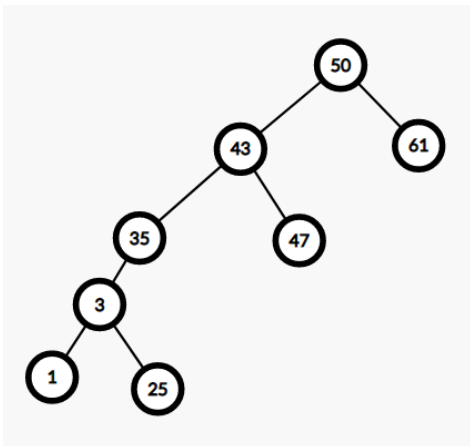


- void **f7()** – Perform preOrder traversal from the root and find the first node p having right son and price in the interval [30,70]. Rotate p to left about its right son.

Content of the file f1.txt:

(AB,50) (EB,43) (G1,35) (A2,3) (A1,1) (B2,25) (X2,47) (BD,61)

(BD,61) (AB,50) (EB,43) (G1,35) (A2,3) (A1,1) (B2,25) (X2,47)

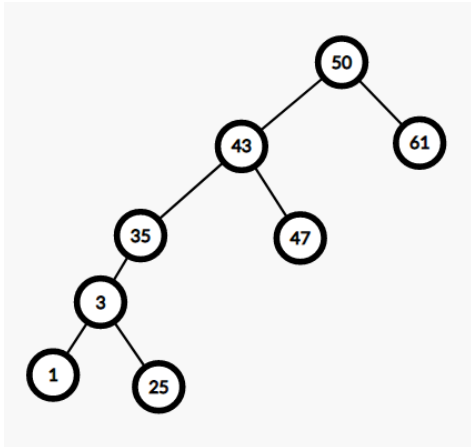


- void **f8()** – Perform preOrder traversal from the root and find the first node p having left son and price in the interval [30,70]. Rotate p to right about its' left son

Content of the file f1.txt:

(AB,50) (EB,43) (G1,35) (A2,3) (A1,1) (B2,25) (X2,47) (BD,61)

(EB,43) (G1,35) (A2,3) (A1,1) (B2,25) (AB,50) (X2,47) (BD,61)

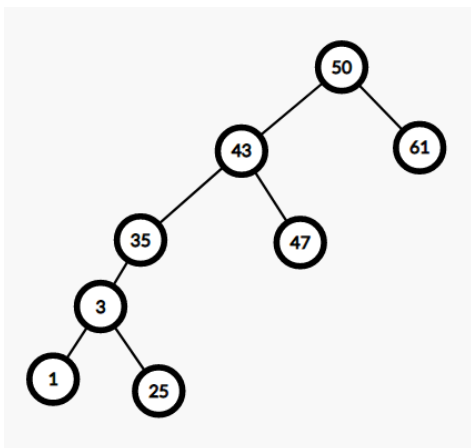


- **void f9():** Suppose p is the 5-th node when performing the in-order traversal of the tree. Calculate the height p and write p's height value to file f1.txt.

Content of the file f1.txt:

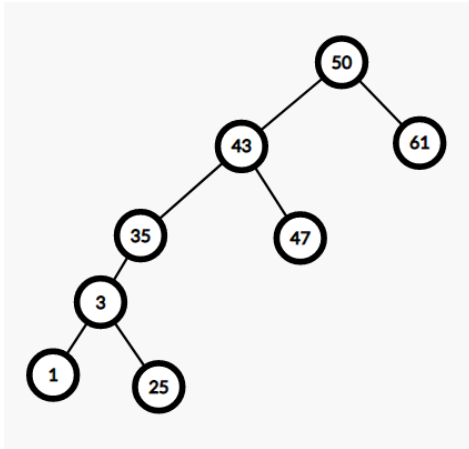
(A1,1) (A2,3) (B2,25) (G1,35) (**EB,43**) (X2,47) (AB,50) (BD,61)

4



void f10() Suppose p is the 5th node when performing the in-order traversal of the tree, count the number of nodes in the subtree with root p and write that value to file f1.txt.

(A1,1) (A2,3) (B2,25) (G1,35) (**EB,43**) (X2,47) (AB,50) (BD,61)



Question 3: (2 marks)

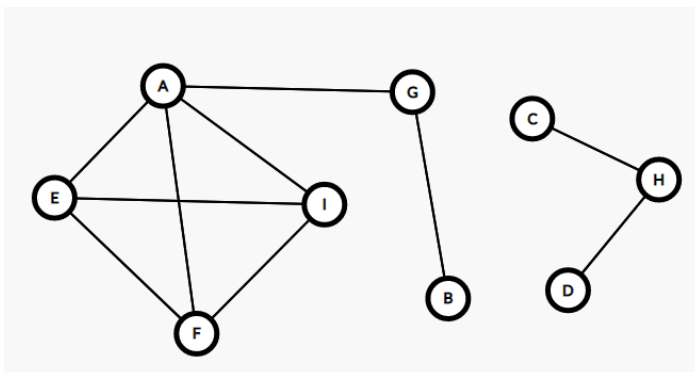
In this question you should complete some methods in **Graph.java** file.

The class Graph is the implementation of a graph. The following methods should be completed:

- void **f1()** - Perform depth-first traversal from the vertex $i=1$ (the vertex B) but display vertices with their degrees in brackets.

B G A E F I C H D

B(1) G(2) A(4) E(3) F(3) I(3) C(1) H(2) D(1)



- void **f2()** - Perform bread-first traversal from the vertex $i=0$ (the vertexA) but display the results in two lines. The first line is vertices, the second line is their degrees correspond.

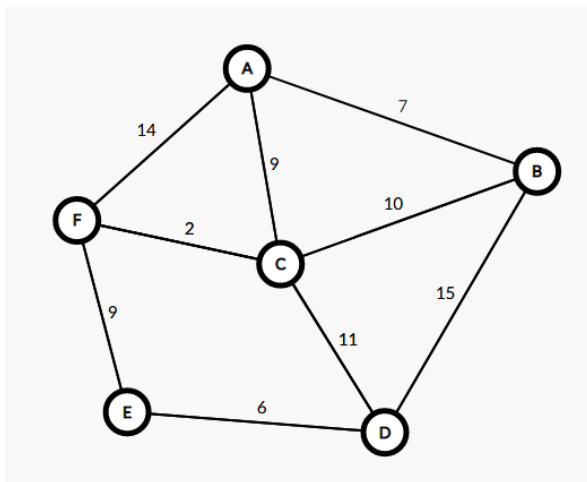
A B F C D E G

2 4 2 2 4 4 2

- void **f3()** - Perform depth-first traversal from the vertex $i=1$ (the vertex B) but display 4 vertices from vertex 3rd to 6th only.

B A C D E F

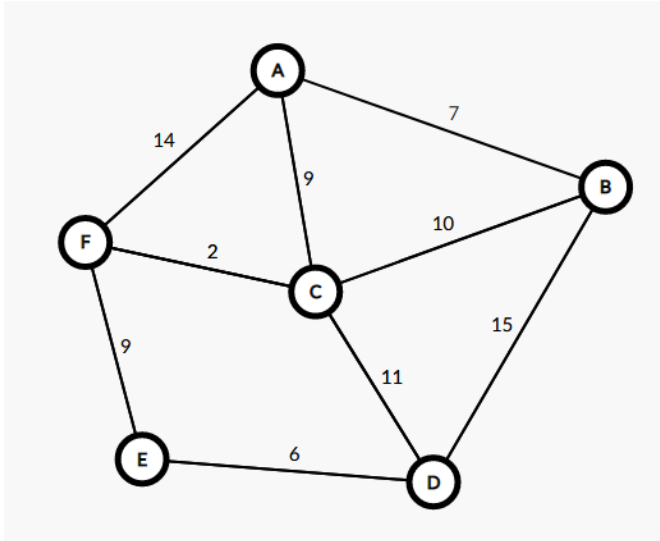
C D E F



- void **f4()** - Perform bread-first traversal from the vertex $i=1$ (the vertex B) but display 3 vertices from the 2nd to the 4th vertex only.

B A C D E F

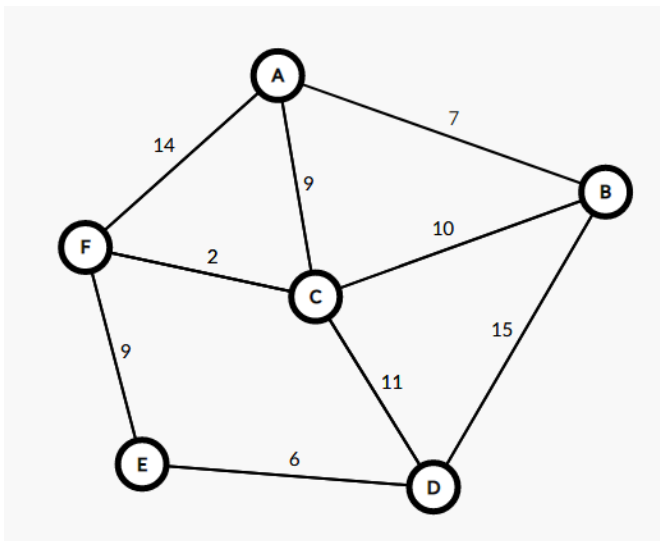
A C D



- void **f5()** – Apply the Dijkstra’s shortest path algorithm to find the shortest path from the vertex 1 (B) to the vertex 4 (E). (Note that in the weighted matrix, the value 999 is considered as infinity). Write 2 lines into the file f1.txt. The first line contains the list of vertices in the shortest path. The second line contains shortest distances to the vertices in the first line.

B C F E

0 10 12 21

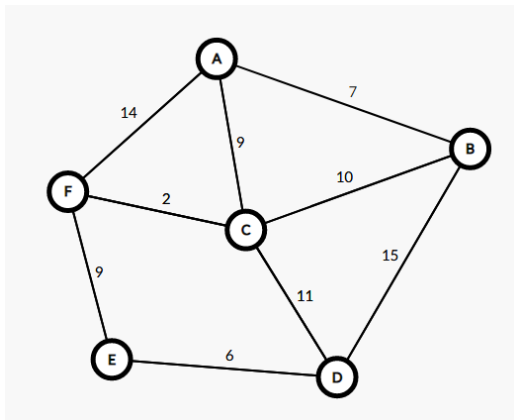


- void **f6()** – Apply the Dijkstra’s shortest path algorithm to find (1) the shortest path from the vertex 1 (B) to the vertex 4 (E), then (2) from 0 (A) to vertex 5(F). (Note that in the weighted matrix, the value 999 is considered as infinity). Write 2 lines

into the file f1.txt. The first line contains the first and the last vertices and shortest distance in (1). Line 2 contains the last 4 vertices selected into the set S with their labels in (2). The output must be following:

B E->21

A-0 C-9 F-11

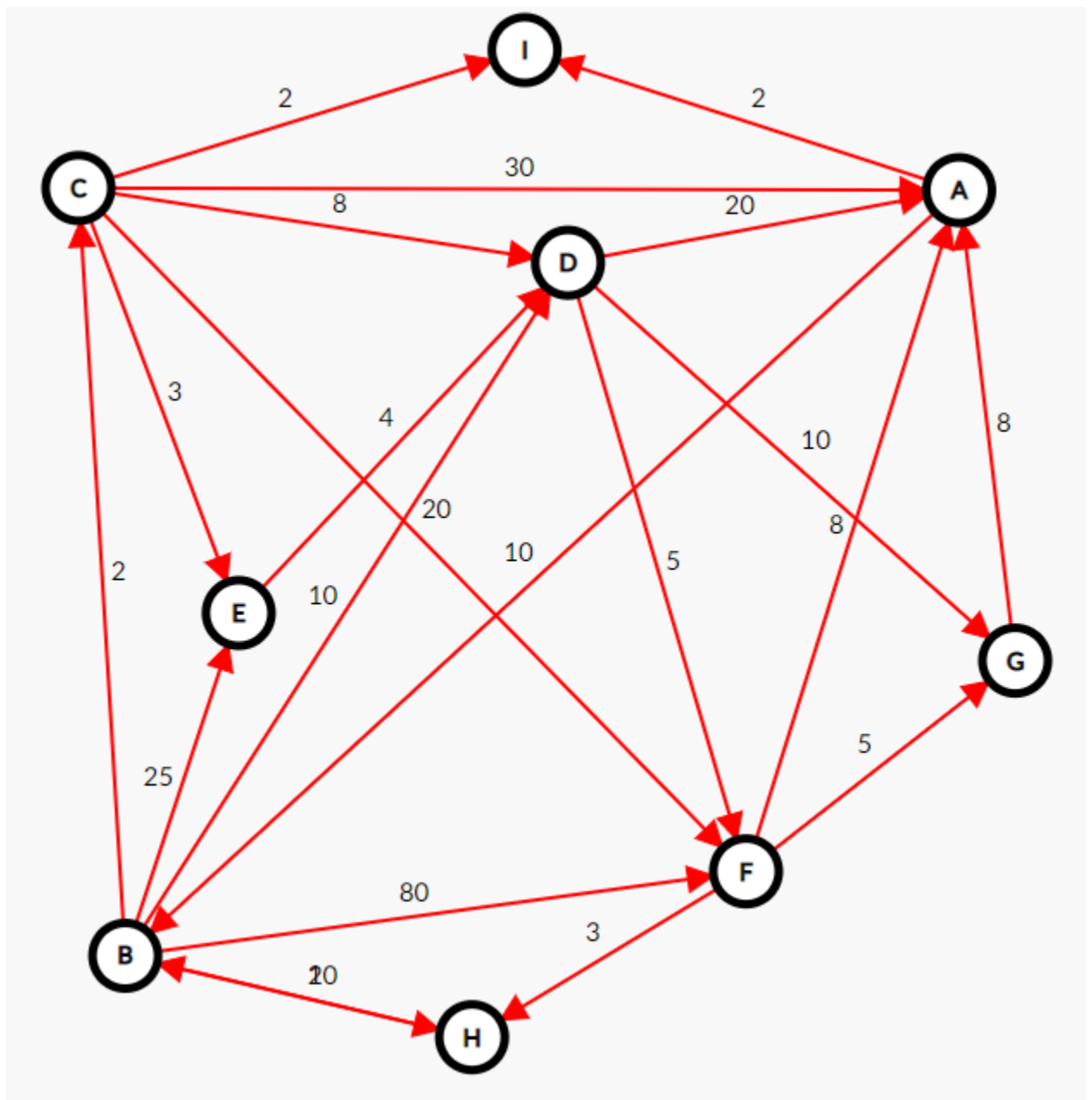


- void **f7()** – Apply the Dijkstra's shortest path algorithm to find (1) the shortest path from the vertex 1 (B) to the vertex 4 (E), then (2) from 1 (B) to vertex 6(G). (Note that in the weighted matrix, the value 999 is considered as infinity). Write 3 lines into the file f1.txt. The first line contains vertices in shortest path (1), line 2 contains the shortest distance in (1), line 3 contains the first 4 vertices selected into the set S with their labels in (2) The output must be following:

B C E

5

E-5 D-9 F-14 G-19



- void **f8()** – Supposed the given graph has Euler's cycle write statements to find the Euler's cycle from the vertex 4 (E). Output in the file **f1.txt**

EDCBDEGFABE

