

Öğrenci Numarası : _____ Adı Soyadı : _____

| | | | | | | |
|------|----|----|----|----|----|--------|
| Soru | 1 | 2 | 3 | 4 | 5 | Toplam |
| Puan | 20 | 20 | 20 | 20 | 20 | 100 |
| Not | | | | | | |

1. (20P) 21, 42, 16, 4, 1, 8 değerlerini boyutu **7** olan seksek özetleme(hopscotch hash) tablosuna en büyük uzaklık değeri(MAX_DIST) **3** olacak şekilde ekleyin. Değerleri silip tekrar yazmamanız için aşağıdaki tablolar verilmiştir.

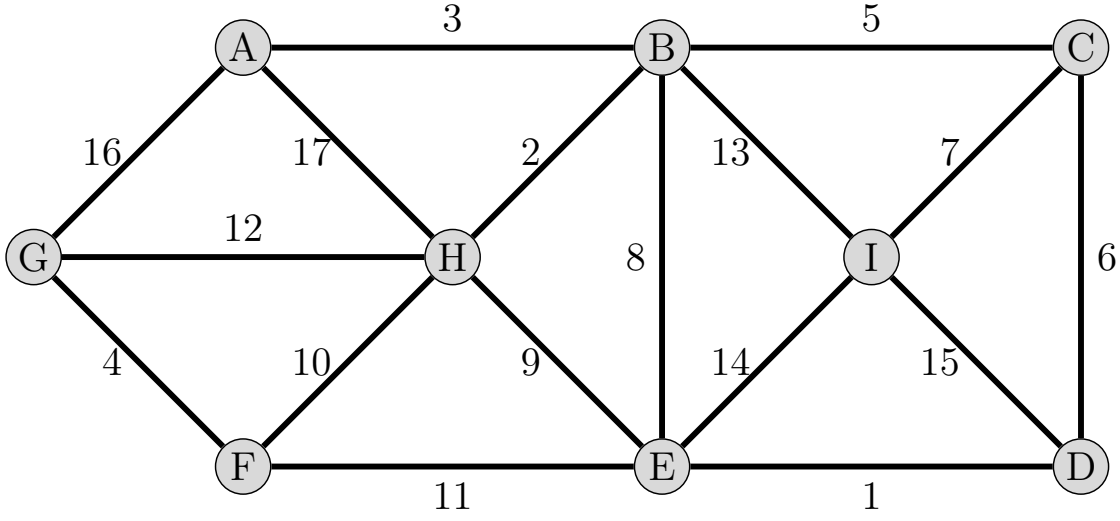
| | Değer | Hop | | Değer | Hop | | Değer | Hop |
|---|-------|-----|---|-------|-----|---|-------|-----|
| 0 | 21 | 110 | 0 | 21 | 110 | 0 | 21 | 110 |
| 1 | 42 | 001 | 1 | 42 | 001 | 1 | 42 | 011 |
| 2 | 16 | 100 | 2 | 16 | 100 | 2 | 8 | 001 |
| 3 | 1 | 000 | 3 | 1 | 000 | 3 | 1 | 000 |
| 4 | 4 | 100 | 4 | | 010 | 4 | 16 | 010 |
| 5 | | | 5 | 4 | 000 | 5 | 4 | 000 |
| 6 | | | 6 | | | 6 | | |

2. (20P) Hızlı sıralama(quick sort) algoritmasındaki bölme(partition) işlemini gerçekleştiren C# kodunu yazınız. Bölme işlemi dizinin belirli bir aralığında olabilmektedir. **left** ve **right** değerleri sırasıyla parçalamanın başlayacağı ve biteceği konumu gösterir. Pivot değeri olarak dizinin **left** indisindeki eleman seçilecektir. `public static void swap(int[] dizi, int i, int j)` metodu dizinin i ve j indisindeki elemanların yerlerini değiştirmeye yarar, partition işlemi içinde kullanabilirsiniz.

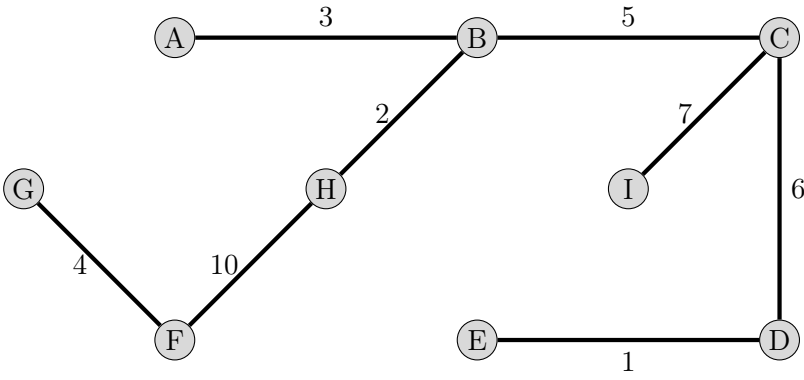
```
public static void partition(int[] dizi, int left, int right)
{
    int pivot = dizi [left];
    int i = left + 1;
    int j = right;

    while (true) {
        while (i <= j && dizi [i] <= pivot)
            i++;
        while (i <= j && dizi [j] >= pivot)
            j--;
        if (i > j)
            break;
        swap (dizi, i, j);
    }
    swap (dizi, left, j);
}
```

3. (20P) Aşağıdaki çizge için Kruskal algoritmasını kullanarak minimum kapsayan ağacı bulun ve toplam maliyeti hesaplayın.



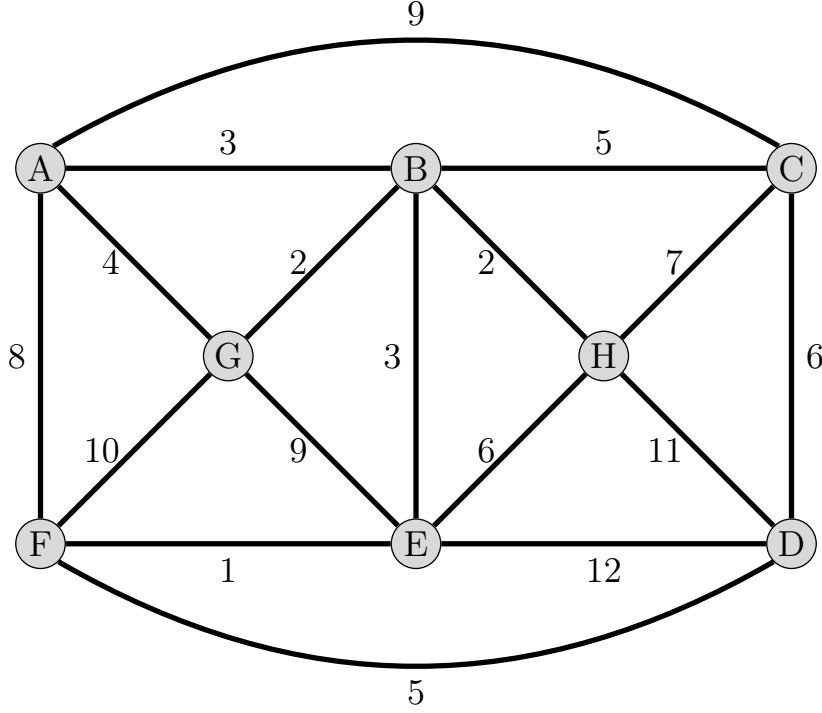
Çözüm:



| Uzunluk | Kenar |
|---------|-----------|
| 1 | ED |
| 2 | BH |
| 3 | AB |
| 4 | GF |
| 5 | BC |
| 6 | CD |
| 7 | IC |
| 8 | BE |
| 9 | EH |
| 10 | HF |
| 11 | EF |
| 12 | GH |
| 13 | BI |
| 14 | EI |
| 15 | ID |
| 16 | AG |
| 17 | AH |

Toplam maliyet: **38**

4. (20P) Aşağıda verilen çizge için Dijkstra'nın en kısa yol algoritmasını kullanarak A düğümünden diğer bütün düğümlere olan en kısa yolları bulun. Tablo içindeki .Y ile biten sütunlar yolları, .U ile biten sütunlar ise yol uzunluklarını belirtir. Örneğin B.Y A'dan B'ye olan yolu, B.U ise o yolun uzunluğunu gösterir.



Tablo 1: Dijkstra Çözüm Tablosu

| Adım | Küme | B.U | B.Y | C.U | C.Y | D.U | D.Y | E.U | E.Y | F.U | F.Y | G.U | G.Y | H.U | H.Y |
|------|------|----------|-----|----------|-----|----------|-------|----------|-----|----------|------|----------|-----|----------|-----|
| 1 | A | 3 | AB | 9 | AC | ∞ | - | ∞ | - | 8 | AF | 4 | AG | ∞ | - |
| 2 | AB | | | 8 | ABC | ∞ | - | 6 | ABE | 8 | AF | 4 | AG | 5 | ABH |
| 3 | ABG | | | 8 | ABC | ∞ | - | 6 | ABE | 8 | AF | | | 5 | ABH |
| 4 | A..H | | | 8 | ABC | 16 | ABHD | 6 | ABE | 8 | AF | | | | |
| 5 | A..E | | | 8 | ABC | 16 | ABHD | | | 7 | ABEF | | | | |
| 6 | A..F | | | 8 | ABC | 12 | ABEFD | | | | | | | | |
| 7 | A..C | | | | | 12 | ABEFD | | | | | | | | |

5. (20P) Araya ekleme sıralamayı gerçekleştiren `public static void insertionSort(int[] dizi)` C# metodunu yazınız.

```
public static void insertionSort(int[] d)
{
    for (int i = 1; i < d.Length; i++) {
        int deger = d [i];
        int j=i-1;
        while(j>=0 && d[j]>deger) {
            d [j + 1] = d [j];
            j = j - 1;
        }
        d [j+1] = deger;
    }
}
```