

CENG 215 Veri Yapıları Laboratuvarı

Ödev 4: Çizge İşlemleri

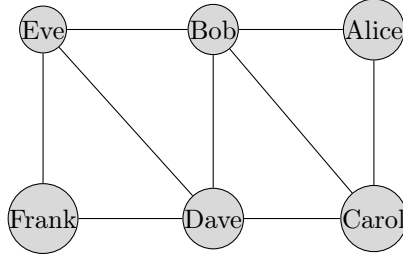
Şevket Umut ÇAKIR

29 Aralık 2023

İnsanların tanıdığı ya da arkadaş olduğu diğer insanları saklayacak bir sosyal ağ yapısı oluşturulmak istenmektedir. Bunun için bir graf temsilinin uygun olacağı düşünüldükçe SocialNetwork sınıfı oluşturulmuştur.

1 SocialNetwork Sınıfı

Verilen SocialNetwork sınıfı bir sosyal ağı temsil etmektedir. Sınıfın içinde bir kişinin ismini arkadaş listesine eşleyen bir Map türünde değişken bulunmaktadır. Sınıfın yapıcı metodu adjacencyList değişkenini HashMap türünde bir değişkene başlatır.



Şekil 1: Örnek çizge

Sınıf içinde aşağıdaki metotlar bulunmaktadır:

- `addPerson(String name)`: Sosyal ağa bir kişi ekler. Eğer kişi daha önceden eklenmemişse, `adjacencyList` değişkenine yeni bir `ArrayList` eklenir. **Bu metodu değiştirmeyin.**
- `addFriendship(String person1, String person2)`: İki kişi arasında bir arkadaşlık ekler. Eğer kişiler daha önceden eklenmemişse, `addPerson` metodu ile eklenir ve `adjacencyList` değişkenine kişilerin arkadaş listeleri eklenir. **Bu metodu değiştirmeyin.**
- `toString()`: Graftaki köşe (vertex) ve kenarların (edge) string temsilini döndürür. **Bu metodu değiştirmeyin.**

Örnek çıktı:

Vertices:

Alice
Bob
Carol
Dave
Eve
Frank

Edges:

Alice: [Bob, Carol]
Bob: [Alice, Carol, Dave, Eve]
Carol: [Alice, Bob, Dave]
Dave: [Bob, Carol, Eve, Frank]
Eve: [Bob, Dave, Frank]
Frank: [Dave, Eve]

- `public int mostFriendCount()`: Sosyal ağda en fazla arkadaşı bulunan kişinin arkadaş sayısının bulunması istenmektedir. Şekil 1'de verilen örnek çizge için bu sayı dördüttür. Dave ve Bob'un arkadaş sayısı dördüttür.
- `public List<String> recommendedFriends(String person)`: Önerilen arkadaş listesini hesaplayıp döndüren metottur. Önerilen arkadaş listesi bir kişinin arkadaşlarının arkadaş kümesidir ve kişinin kendi arkadaşlarını içermez. Şekil 1'deki örnek için Alice'in önerilen arkadaş listesi [Dave, Eve] olacaktır. Eve için öneri listesi [Alice, Carol] olacaktır. Bu metodun verimli çalışması için `java.util` paketinde bulunan `Set` türünü kullanabilirsiniz.

Listing 1: SocialNetwork.java

```

1 import java.util.ArrayList;
2 import java.util.HashMap;
3 import java.util.List;
4 import java.util.Map;
5
6 /**
7  * Verilen SocialNetwork sınıfı bir sosyal ağı temsil etmektedir. Sınıfın içinde bir
8  * ↪ kişinin ismini
9  * arkadaş listesine eşleyen bir Map türünde değişken bulunmaktadır.
10 * Bu sınıfın sadece mostFriendCount ve recommendedFriends metotlarını değiştirin.
11 * Diğer metot ve özellikler testlerin çalışması için gereklidir.
12 */
13 public class SocialNetwork {
14     // ***** Lütfen bu aralıktaki kodları değiştirmeyin
15     ↪ *****
16     // Her bir kişinin ismini arkadaş listesine eşleyen bir harita
17     private Map<String, List<String>> adjacencyList;
18     // Map nesnesi getter
19     public Map<String, List<String>> getAdjacencyList() {
20         return adjacencyList;
21     }
22
23     // Yapıcı
24     public SocialNetwork() {
25         adjacencyList = new HashMap<>();
26     }
27
28     // Sosyal ağa bir kişi ekler
29     public void addPerson(String name) {
30         if (!adjacencyList.containsKey(name)) {
31             adjacencyList.put(name, new ArrayList<String>());
32         }
33     }
34
35     // İki kişi arasında bir arkadaşlık ekler
36     public void addFriendship(String person1, String person2) {
37         addPerson(person1);
38         addPerson(person2);
39         if (!adjacencyList.get(person1).contains(person2)) {
40             adjacencyList.get(person1).add(person2);
41             adjacencyList.get(person2).add(person1);
42         }
43     }
44
45     // Graftaki köşe(vertex) ve kenarların(edge) string temsilini döndürür
46     @Override
47     public String toString() {
48         StringBuilder builder = new StringBuilder();

```

```

47     builder.append("Vertices:\n");
48     for (Map.Entry<String, List<String>> entry : adjacencyList.entrySet()) {
49         String person = entry.getKey();
50         builder.append(person + "\n");
51     }
52     builder.append("\nEdges:\n");
53     for (Map.Entry<String, List<String>> entry : adjacencyList.entrySet()) {
54         String person = entry.getKey();
55         List<String> friends = entry.getValue();
56         builder.append(person + ": ["");
57         for (int i = 0; i < friends.size(); i++) {
58             String friend = friends.get(i);
59             if (i == friends.size() - 1) {
60                 builder.append(friend);
61             } else {
62                 builder.append(friend + ", ");
63             }
64         }
65         builder.append("]\n");
66     }
67     return builder.toString();
68 }
69 // ***** Lütfen bu aralıktaki kodları değiştirmeyin
70 ↪ *****
71 /**
72  * Grafta en çok arkadaşı olan kişinin arkadaş sayısını döndürür
73  * @return En çok arkadaş sayısı
74  */
75 public int mostFriendCount() {
76     return -1;
77 }
78
79 /**
80  * Parametre olarak verilen kişi için önerilen arkadaş listesi
81  * oluşturur. Önerilen arkadaş listesi aşağıdaki şekilde hesaplanır:
82  * A: Parametre olarak verilen kişi
83  * B: A'nın arkadaş kümesi
84  * C: B'deki kişilerin arkadaşlarının kümesi
85  * Öneri listesi: (B-C)-A
86  * Öneri listesinde her isim sadece 1 defa yer almalıdır, ve listede
87  * kişinin kendi arkadaşları ve kendisi bulunmamalıdır
88  * Lütfen ödev dokümanındaki örneği inceleyiniz.
89  * @param person Parametre olarak verilen kişi
90  * @return Arkadaş listesi
91  */
92 public List<String> recommendedFriends(String person) {
93     List<String> recommended = new ArrayList<>();
94     return recommended;
95 }

```

2 Ödev Açıklamaları

- Ödevler <https://bilmoodle.pau.edu.tr> adresindeki arayüzlerden cevaplanacaktır. Ödevinizdeki kod dosyasını kişisel bilgisayarınıza indirip, ödevi çözüp tekrar sisteme geri yükleyebilirsiniz fakat sınıf, dosya ve paket yapısının korunduğundan emin olmalısınız.
- Ödevler bireysel olarak cevaplanmalıdır, grup çalışması yapmak yasaktır

- Ödevleri yapay zeka sohbet robotlarına cevaplatmak yasaktır
- Ödevlerde kopya kontrolü yapılacaktır ve benzerlik oranı belirli bir yüzdenin üzerinde olan ödevler kopya olarak değerlendirilecektir. Değişken isimlerini değiştirmek, kodların sırasını değiştirmek anlaşılabilen yöntemlerdir, lütfen bu yola başvurmayınız.
- Ödevin son teslim tarihi **07.01.2023 saat 23:59**'dur. Belirtilen saatte sistem kapanacaktır, lütfen son ana kadar beklemeden ödevinizi gönderin.