**DATA STRUCTURES**

**2023-2024 Fall Semester**

**PROJECT 4**

**GRAPHS, GRAPH ALGORITHMS, TREES and OTHER SUBJECTS**

**iii)** **Insert 3** and redraw **Heap** in Fig 2. (Şekildeki **Yığına (Heap)** **3 değerini** ekleyerek yeniden çiziniz).

**iv)** After that, remove a value from heap and redraw it (Ardından yani 3 değeri eklendikten sonraki durumda yığından eleman silerek yeniden çiziniz).

**i) Insert 3** and **redraw AVL Tree** in Figure 1 (Şekildeki **AVL Ağacına** **3 değerini** ekleyerek yeniden çiziniz).

**ii)** After that, **insert 5** and **redraw** also new **AVL Tree** (Ardından **5 değeri ekleyerek** yeniden **çiziniz**). Ağaç 3 eklenmiş halde olacak!

**1)**

**Fig. 2**

**(10 points)**

**(10 points)**

**Fig. 1**

1. **Learn and Write a C # (or Java) method for B-Tree Insertion and test with your own values. Explain the steps of the method. (**C# (veya Java) dilinde B-Tree ekleme metodunu **öğreniniz**, **kodlayıp** çalıştırınız ve kendi değerlerinizle **test ediniz.** Yapılan **işlemleri yazarak anlatınız.**). **(10)**

[Dileyenler B-Tree yerine, İnternet’ten veya diğer kaynaklardan araştırarak ve hazır kodlardan da yararlanarak aşağıdaki alternatifleri de tercih edebilirler. Alıntı yaptığınız kaynaklara atıf verin.]

**A1)** **AVL-Tree** Insertion method (sadece ekleme metodu)

**A2)** **Red-Black Trees** (Textbook Ch. 9) ***or* Huffman Encoding Tree** (Textbook 415-421)

1. **Read Chapters 13 and 14 from the textbook. Code and test the following methods for Weighted Graphs in C # / Java / C** (Ders kitabı Bölüm 13 ve Bölüm 14’ü okuyunuz. C# / Java / C ‘de Ağırlıklı Çizgeler için aşağıdaki metotları kodlayınız ve test ediniz) **(15)**:
   1. **Dijkstra’s Shortest Path** [Derste anlatılan koda dersin sayfasından ulaşabilirsiniz],
   2. Prim’s MST (Minimum Spanning Tree) [Ders kitabında var]
   3. BFT(Breadth-First Traverse) **or** DFT (Depth-First Traverse)
   4. Fill in the table below by understanding and learning from the textbook (aşağıdaki tabloyu ders kitabından araştırarak öğreniniz ve doldurunuz):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Dijkstra’s SP** | Prim’s MST | BFT | **Heap Deletion** |
| **Big-O** (Zaman Karmaşıklığı  Big-O Notasyonuna Göre) |  |  |  |  |

**4) a)** Şekil’de gösterilen **Sosyal Ağı, Graph (Çizge) yapısı** şeklinde **Python dilinde NetworkX kütüphanesi kullanarak oluşturunuz** ve **görsel olarak çizdiriniz**.

**b) Her kişi çiftinin birbirlerinden uzaklıklarını hazır** dijkstra\_path **fonksiyonu ile hesaplatınız.** <http://avinashu.com/tutorial/pythontutorial/NetworkXBasics.html>, <https://www.geeksforgeeks.org/directed-graphs-multigraphs-and-visualization-in-networkx/> [for digraphs]

**c) Verilen bir köşe numarasından başlatarak hazır fonksiyonlarla DFS ve BFS dolaşınız.**

Çizgenin kenarlarındaki ağırlıklar, kişilerin birbirlerinin çağrılarına ne kadar süre sonra cevap verdiklerini ifade etmektedir. Bağlantı olmayan kişilerin birbirlerini doğrudan tanımadıklarını varsayabilirsiniz.

**d) Bu örnek için, SP, BFS ve MST bulmanın ne işe yarayabileceğini ve hangi amaçlarla kullanılabileceğini raporda kısaca belirtiniz.**

**e)** Bu tür bir çizgede, verilen bir kişinin en kısa sürede ulaşabileceği doktoru nasıl bulursunuz? (ve kaç arama gerekeceğini). **Etkin bir yöntem adı verip algoritmasını yazınız. Karmaşıklığını belirtiniz. Aranan her kişinin belli bir yüzde ile telefonu aç(a)mayacağını düşünecek olursanız yönteminizi nasıl değiştirmeniz gerekeceğini belirtiniz.**

diyagram, çizgi, metin, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. i) Compare **Prim’s Algorithm** and **Kruskal’s Algorithm** briefly: **(5)**

[Prim’in ve Kruskal’ın MST algoritmalarını kısaca karşılaştırınız]

ii) Implement a Trie Data Structure and also write the insert (to the trie) method in C# or Java. Test if it works correctly. Explain the code in your own words briefly. (Trie veri yapısı düğümü ve eleman ekleme metodunu kendiniz yazınız veya hazır kod bularak yazınız. Çalıştırıp doğruluğunu kontrol ediniz. Kodu kendi cümlelerinizle kısaca anlatınız.) **(5)**

iii) Explain only 4 of the terms below (Aşağıdaki kavramlardan istediğiniz 4 tanesini seçerek birer paragraf kurşunkalemle yazarak tanımlayınız. Birçoğu derste anlatılmaktadır, ders kitabında da bulunmaktadır) **(15)**:

* 1. B+Tree
  2. 2-3-4 Tree

c. Quadratic Probing

* 1. Warshall’s Algorithm
  2. k-d Tree
  3. Topological Sorting
  4. Dynamic Programming
  5. R-Tree

**Self-assessment Table**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Points** | **Estimated Grade** | **Explanation** |
| **1 a) AVL Tree** | **10** |  |  |
| **1 b) Heap** | **10** |  |  |
| **2) B-Tree Insertion / AVL Tree Insertion / Red-Black Trees / Huffman Encoding Tree** | **10** |  |  |
| **3 a) Dijkstra’s shortest path code + test** | **4** |  |  |
| **3 b) Prim’s MST code + test** | **4** |  |  |
| **3 c) BFT or DFT code + test** | **3** |  |  |
| **3 d) Filling Big-O Table** | **4** |  |  |
| **4 a) Graph Drawing** | **3** |  |  |
| **4 b) Finding Shortest Paths with Dijkstra’s** | **3** |  |  |
| **4 c) DFS and BFS** | **3** |  |  |
| **4 d) Thinking and Writing aims of given algorithms** | **3** |  |  |
| **4 e) Real Life Application** | **3** |  |  |
| **5 i) Comparison (Prim’s & Kruskal’s Algorithm)** | **5** |  |  |
| **5 ii) Trie Data Structure and Insertion Method** | **5** |  |  |
| **5 iii) Explanations of 4 terms** | **10** |  |  |
| **Demo Video for Source Codes and Tests of Q2 Q3a, Q3b, Q3c and Q5.ii .** | **10** |  |  |
| **Self-assessment Table** | **10** |  |  |
| **Total** | **100** |  |  |
|  |  |  |  |

**Not:** Özdeğerlendirme Tablosunda her bir maddeden alınması beklenen puanı yazınız ve yapmadığınız alt maddeleri belirtiniz**.**

Proje 4’ü 4 kişi ortak veya tek kişi teslim edebilirsiniz.

Kodlama sorularında, puan kırılmaması için ilgili soruda belirtilen programlama dillerden birisinde kodlayınız.

Pandemi öncesine göre 1 soru azaltılmış; Diğer bir soruda kurşunkalemle algoritma yazma işlemi kaldırılmıştır.