**DATA STRUCTURES**

**2021-2022 Fall Semester**

**PROJECT 4**

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

**iii)** **Insert 1** and redraw **Heap** in Fig 2. (Şekildeki **Yığına (Heap)** **1 değerini** ekleyerek yeniden çiziniz).

**iv)** After that, remove a value from heap and redraw it (Ardından yani 1 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 2** and **redraw** also new **AVL Tree** (Ardından **2 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) **(20)**:
   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 Insertion** |
| **Big-O** (Zaman Karmaşıklığı  Big-O Notasyonuna Göre) | O(ElogV) | O(ElogE) | O(|V|+|E|) | O(log n) |



1. **(15)**
2. Python NetworkX kütüphanesini / eşdeğerini kullanarak **yandaki şekilde** verilen ağırlıklı çizgeyi oluşturup grafiksel olarak çizdiriniz. İpucu: <http://avinashu.com/tutorial/pythontutorial/NetworkXBasics.html>

<https://www.geeksforgeeks.org/directed-graphs-multigraphs-and-visualization-in-networkx/> [for digraphs]

1. 4 numaralı düğümden tüm diğer düğümlere en kısa yol uzunluklarını buldurunuz. Hazır dijkstra\_path komutunu kullanınız.
2. 1 numaralı düğümü siliniz ve i) adımını tekrarlayınız.
3. a) Compare **Prim’s Algorithm** and **Kruskal’s Algorithm** briefly: **(5)**

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

b) Explain only 5 of the terms below (Aşağıdaki kavramlardan istediğiniz 5 tanesini seçerek birer paragraf kurşunkalemle yazarak tanımlayınız. Birçoğu derste anlatılmaktadır, ders kitabında da bulunmaktadır) **(20)**:

* 1. B+Tree
  2. 2-3-4 Tree
  3. Trie
  4. Quadratic Probing
  5. Warshall’s Algorithm
  6. k-d Tree
  7. Topological Sorting
  8. Dynamic Programming
  9. 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** | **5** |  |  |
| **3 b) Prim’s MST code + test** | **5** |  |  |
| **3 c) BFT or DFT code + test** | **5** |  |  |
| **3 d) Filling Big-O Table** | **5** |  |  |
| **4 i) Graph Drawing** | **5** |  |  |
| **4 ii) Finding Shortest Paths with Dijkstra’s** | **5** |  |  |
| **4 iii) Node deletion and repeating i, ii.** | **5** |  |  |
| **5 a) Comparison (Prim’s & Kruskal’s Algorithm)** | **5** |  |  |
| **5 b) Explanations of 4 terms** | **20** |  |  |
| **Demo Video for Source Codes and Tests of Q2 Q3a, Q3b and Q3c.** | **5** |  |  |
| **Self-assessment Table** | **5** |  |  |
| **Total** | **100** |  |  |
|  |  |  |  |

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

Proje 4’ü 2 kişi ortak veya tek kişi teslim edebilirsiniz.

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

Geçen seneye göre 1 soru azaltılmış; Diğer bir soruda kurşunkalemle algoritma yazma işlemi kaldırılmıştır.