CSE2025 2nd PROJECT REPORT

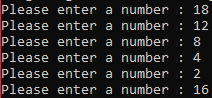
Yasin Enes POLAT 150117015

1-Entering Numbers

User can enter numbers in one line like this;



or can enter a number for each line like this;



In both cases, user must enter a ‘0’ to make program stop getting numbers.

2-BST Created and In-Order Traversed

---IN-ORDER TRAVERSAL---

Data: 2 Depth: 4

Data: 4 Depth: 3

Data: 8 Depth: 2

Data: 12 Depth: 1

Data: 14 Depth: 3

Data: 15 Depth: 4

Data: 16 Depth: 2

Data: 17 Depth: 3

Data: 18 Depth: 0

Data: 20 Depth: 3

Data: 21 Depth: 4

Data: 24 Depth: 2

Data: 36 Depth: 1

Data: 54 Depth: 3

Data: 72 Depth: 2

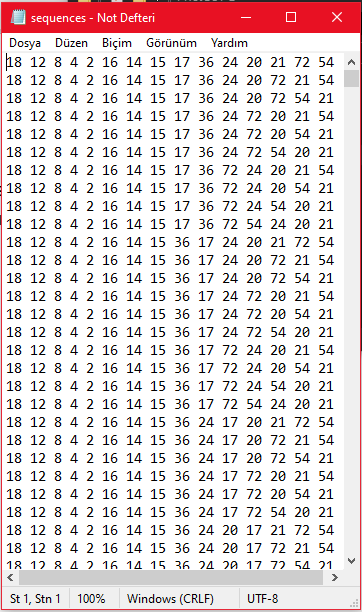
3-My Solution

In my solution, program gets all numbers from user and adds them to the BST respectively. Then traverses the tree again and find all possible nodes can come after the node program is working on, then adds them to the related node’s linked list. When my program starts to print out the sequences, looks for path to the node-x if all nodes that should come before the node-x are in the sequence or not. If all of them are int the sequence yet, it adds node-x to the end of the current sequence.

Benim çözümümde programım bütün sayıları kullanıcıdan alıp sırasıyla ekleyerek ağacı oluşturuyor. Sonrasında bütün ağacı tekrar gezerek her bir node için o nodedan sonra gelebilecek bütün olası nodeları o nodeun içerisindeki listeye ekliyor. Aynı ağacı veren dizileri yazdırırken de bir nodeu diziye eklemeden önce o nodea kadar olan yol üzerindeki nodeların halihazırda dizide var olup olmadığını kontrol ediyor ve eğer tamamı mevcutsa o nodeu da dizinin sonuna ekliyor. Recursive şekilde çalışan fonksiyon global değişken olan total sayısına ulaştığında da oluşturulan diziyi ‘sequences.txt’ dosyasının içerisine yazıyor.

Example sequences that gives same BST topology with the sequence :

18 12 8 4 2 16 14 15 17 36 24 20 21 72 54



Total Sequences : 3,153,150