

(15P) Algoritmanızın zaman karmaşıklığını komşu kovaların yer değiştirme sayısı cinsinden açıklayarak belirleyiniz.

Bu algoritma en fazla $n-1$ yer değiştirme yapacaktır, çünkü son kovalar bir sonraki dolu kova olmayacaktır. Her yer değiştirmede, yalnızca iki kova yer değiştirilir, zaman karmaşıklığı bu yüzden toplam yer değiştirmeye bağlı. En kötü durumda, tüm boş kovalar ve tüm dolu kovalar solda, ve boş kovalar sağda olduğunda $n-1$ yer değiştirme gerekir. Bu yüzden $O(n)$ 'dir.

9) Verilen çizgede DFS (Depth First Search) ağacı çiziniz. Vertex'lerin stack'a eklenme ve çıkma zamanlarını belirtiniz. (Gezmeye Öğrenci numaranızın birler basamağı ile aynı numaralı düğümden başlarız. 0 ise 10'dan başlarız. Komşuluklar küçükten büyüğe sıralıdır.)