

ALGORİTMA ANALİZİ

4.HAFTA ÖDEVİ

HEAP VERİ YAPISI VE KARMAŞIKLIK ANALİZİ

BEDİRHAN ÇELİK - 16040142

1. Heap yapısının diğer isimleri nelerdir?

Heap yapısının diğer ismi "Priority Queues(Öncelik Kuyrukları)" dır.

2. Heap veri yapısının iki özelliği nedir, nasıl avantaj sağlar?

Heap veri yapısının iki özelliği; *Yapı Özelliği (Structure Property)* ve *Yığın Sırası Özelliği (Heap-Order Property)* dir.

- **Yapı Özelliği (Structure Property) :**

Bir yığın, yapı olarak " tam ikili ağaç (complete binary tree)" tır. Ağacın en alt düzeyi hariç, tüm düzeylerinin düğümlerle dolu olması gerekir. En son düzeyde olabilecek boşluklar ise ağacın tamamen sağ tarafında olmalı yani ağaç doldurulmaya hep sol taraftan başlanılmalıdır.

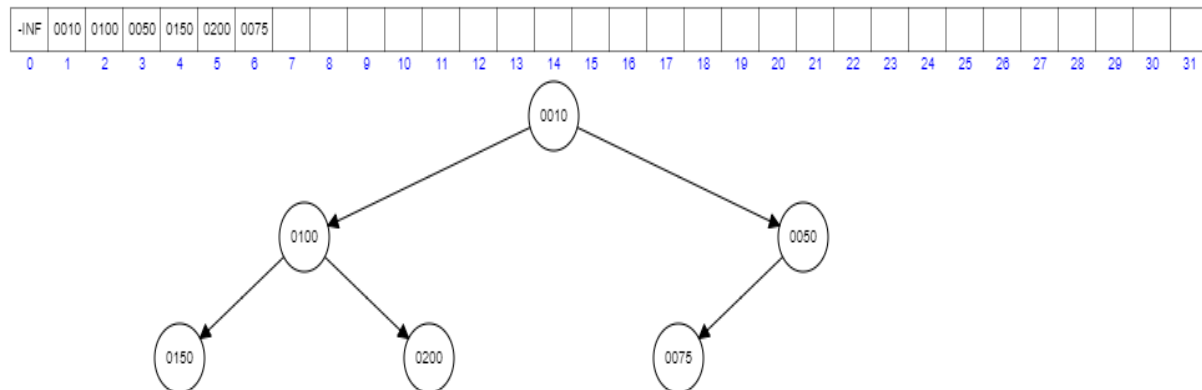
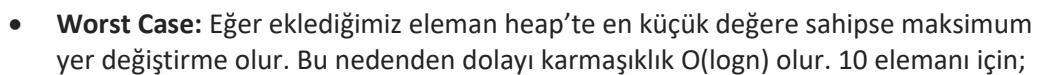
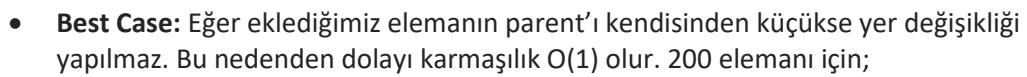
- **Yığın Sırası Özelliği (Heap-Order Property):**

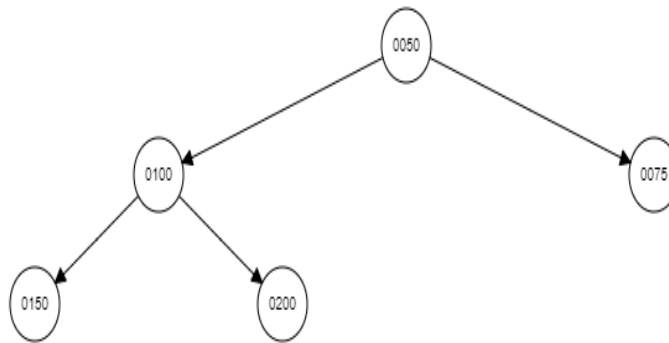
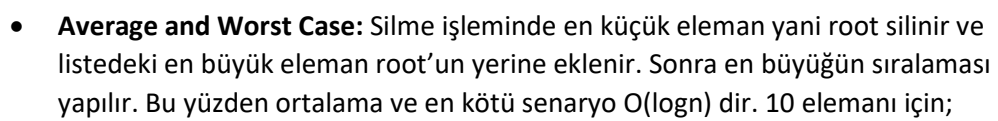
Bir düğüm her zaman çocuklarından küçük bir değere sahip olmalıdır(minHeap için). Böylece kök düğümün ağaçtaki en küçük düğüm olduğunu garanti altına almış oluruz.

3. MaxHeap yapısı ile MinHeap yapısı arasında ne fark vardır?

MaxHeap ve MinHeap arasındaki fark, MinHeap'in küçükten büyüğe, MaxHeap'in ise büyüktan küçüğe sıralamasıdır. Yani eşitlik kontrolü.

MinHeap için ekleme (insert) ve çıkarma (remove,extract) işlemleri uygularsak.
Aşağıdaki heap yapısı için **EKLEME**;



[illegible]

Heap yapısında her bir operasyonun karmaşıklığına bakılırsa (amortized cost) ekleme işleminde sabit bir zaman harcadığı görülür. Her eleman yapının en sonuna eklendiğinden $O(1)$ zaman harcanır. Ancak bazı durumlarda eklenen elemanın parent'ı kendisinden büyüğe yer değiştirme uygulanır. Böylelikle karmaşıklık $O(\log n)$ olur veya yaklaşır. Çıkarma işleminde ise neredeyse her zaman $O(\log n)$ karmaşıklığı görülür. Çünkü silinen elemanın yerine yapıdaki en büyük eleman yerleştirilir ve onun tekrar sıralanması gerekir. Yine bazı durumlarda mesela yapıda 2 eleman varsa silme işlemi $O(1)$ olur.