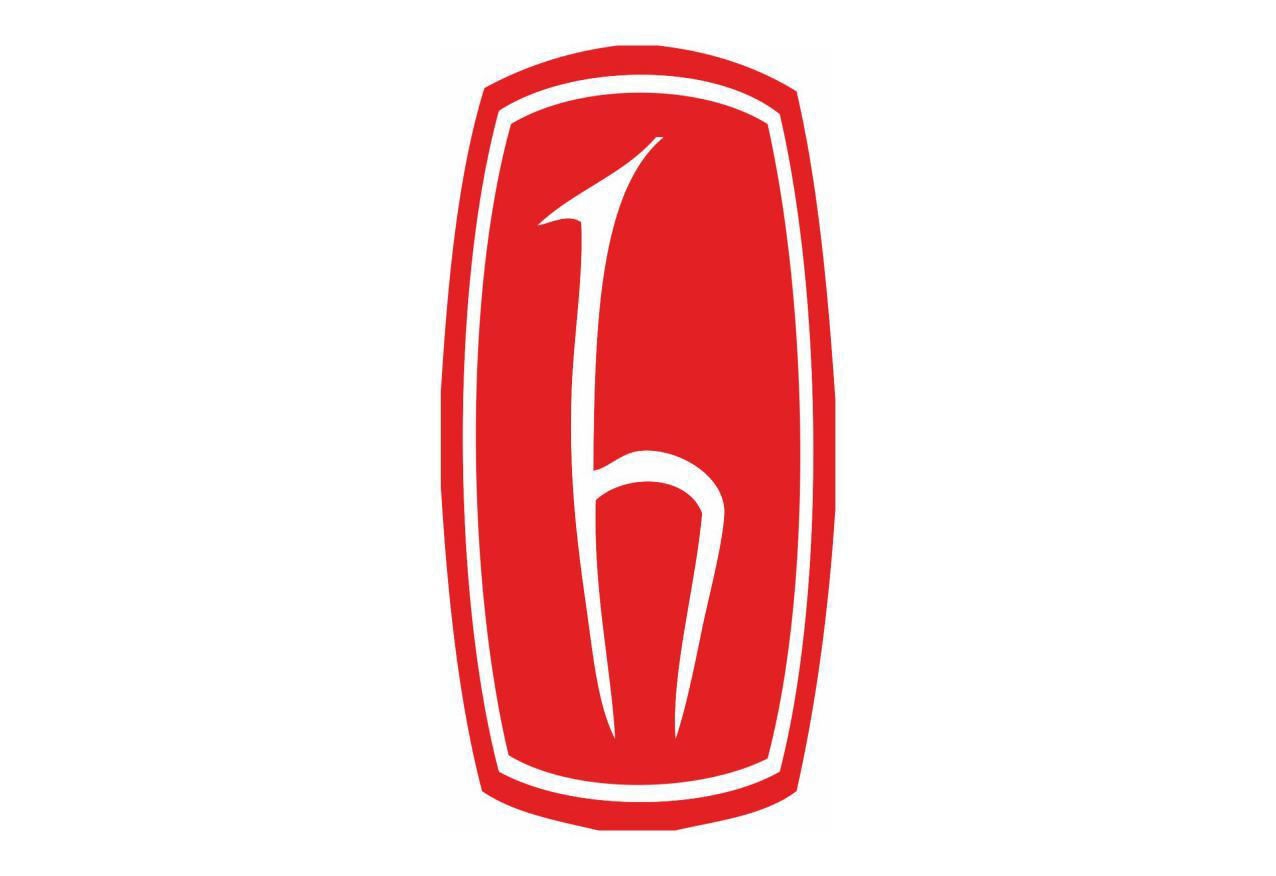
HACETTEPE UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING



**NAME-SURNAME:** DOĞUKAN BERAT KARATAŞ

**NUMBER:** 21527142

**LECTURE:** SOFTWARE LABORATORY II (BBM 204)

**SOFTWARE USAGE:**

1. **Genel Olarak Dosya Okuma, Liste Oluşumu ve Tanımlar:**

Dosya da ilk 3 satırda belli tanımlamalar mevcut;

* İlk satır kullanılacak memory’nin boyutunu,
* İkinci satır hangi “page replacement” algoritmasının kullanılacağını,
* Üçüncü satır da arama algoritmasının yalnızca binary search tree olduğunu bize vermekte.

**Classlar;**

* *Assignment2.java ;* genel işlemlerin yapıldığı class.
* *MemoryList.java ;* Memory’nin tutulduğu class, aynı zamanda ekrana yazdırılmak üzere toString() metodu kullanmaktadır.
* *Queue.java ;* Fifo’da kullanılan queue class’ı.
* *BST.java ;* Search için oluşturulmuş Binary Search Tree class’ı.
* *Heap.java ;* Priority Queue algoritmasında kullanılan class.
* *SecondChance.java ;* Second Chance algoritmasında kullanılan class.

1. **FIFO Page Replacement:**

Queue kullanılarak ilk giren ilk çıkar mantığıyla oluşturulmuş algoritma. Çalışma mantığına bakacak olursak;

* İlk bakılan durum; queue’nun dolu olup, olmaması.
* Eğer memory boş ve eleman halihazırda eklenmemiş ise;

Memory dolana kadar, elemanı hem queue, hem listeye, hem de binary search tree’ye ekle.

* Eğer memory dolu ve eleman halihazırda eklenmemiş ise;

Queue’ya ilk giren eleman “dequeue” edilir ve yeni eleman “enqueue” ile queue’ya eklenir. Bu sırada da listeden aynı eleman yenisi ile değiştirilmiş olur. Aynı zamanda da search algoritması yanlış sonuç vermesin diye, queue’dan çıkarılan eleman; binary search tree’den de çıkarılır ve yeni eleman tree’ye eklenir. Toplamda 3 silme ve 3 ekleme işlemi yapılmış olur.

* Son durum memory dolu da olsa, boş da olsa; eklenecek eleman memory’de varsa işlem yapmadan devam et.

1. **Priority Queue Page Replacement:**

Max Heap Tree yardımıyla, eklenecek elemanın listedeki maximum elemanla yer değiştirdiği algoritma çeşidi. Çalışma mantığına bakacak olursa;

* İlk bakılan durum; memor’nin dolu olup, olmaması
* Eğer memory boş ve eleman halihazırda eklenmemiş ise;

Memory dolana kadar, elemanı hem heapList’e, hem listeye ve hem de binary search tree’ye ekle.

* Eğer memory dolu ve eleman halihazırda eklenmemiş ise;

HeapList oluşturulan Heap class’ında sortlayıp sonrasında max() fonksiyonu ile maximum elemana ulaşıyor. Bu eleman binary search tree’den çıkarılır ve yeni eleman tree’ye eklenir. Aynı zamanda hem heapList’te hem de memoryList’te maximum eleman ile eklenecek eleman yer değiştirilir. Böylelikle her satırda istenilen sıra elde edilmiş olur.

1. **Second Chance Page Replacement:**