BİL212/013

Veri Yapılari

Lab7

Son gönderim tarihi: 12.07.2024 23:59

**Soru-1)** Aşağıdaki soruları kısaca cevaplandırın.

1. “İyi” bir hash fonksiyonu hangi özellikleri sağlamalıdır?
2. Açık adresleme (Open Addressing) ile çakışmalarin (collision) çözüldüğü bir HashTable düşünelim. Load factor (, n HashTable yapısına eklenen elemanların sayısı ve N kapasitesidir, 1’den büyük olabilir mi? Neden olabilir? veya neden olamaz?
3. Linear Probing veya Separate Chaining ile collision çözümleyen bir Hash Table yapısının *put* metodu en kötü durumda (worst case scenario) *O(n)* zaman karmaşıklığına sahiptir. Her iki collision çözümleme yaklaşımı için *put* metodunun O(n) zaman karmaşıklığında çalıştığı birer durumu örnek veriniz.

**Soru-2)** Hash fonksiyonu *h(x) = x* olan bir HashTable (H) düşünelim. S={x | 0 <= x <=m }, m sonsuz olmayan bir tam sayı ve S sonlu bir tam sayı kümesi olsun. Örneğin S={1,2,3,6} ise m=6 olabilir. Verilen S kümesi bir HashTable yapısında tutulmak isteniyor. İlk olarak m boyutunda bir bucket array oluşturulur ve S kümesinin elemanları hash fonksiyonu yardimiyla bu array’e **doğrudan** yerleştirilir. Yapılan işlemler boyunca array’in boyutunun değişmediğini varsayabilirsiniz. Bu şekilde oluşturulan HashTable yapısının görünümü aşağıdaki gibidir:

| null | 1 | 2 | 3 | null | null | 6 |
| --- | --- | --- | --- | --- | --- | --- |

i=0 i=1 i=2 i=3 i=4 i=5 i=6

Bu yapıda aşağıdaki metotlar tanımlıdır:

* put(j), j elemanını hash tablosuna ekler, (H[h(j)]=j)
* delete(j), j elamanını hash tablosundan siler, (H[h(j)]=null)
* exists(j), j elemanının hash tablosunda olup olmadığını kontrol eder (if H[h(j)] != null)

Aşağıdaki soruları cevaplayın. **Herhangi bir kod/pseudo kod yazmanıza gerek yok. Çözümünüzü açıklamanız yeterli.**

1. Bu tarz bir adresleme yapan HashTable yapısının dezavantajı nedir?

1. Bu yapının daha az yer kaplaması için Hash fonksiyonunu değiştirmeden yukarida verilen HashTable yapısında nasıl bir değişiklik yapılmalıdır? Yapacağınız değişiklikler put delete ve exists metotlarının zaman karmaşıklığını **değiştirmemelidir.** (**İpucu:** Elemanların tutulduğu veriyapısı üzerinde değişiklikler yapın)

**Soru-3)** Quadratic Probing ile collision çözümlemesi yapan bir HashTable put get ve delete işlemlerinde elemanların tutulduğu arrayde indeks bulmak için aşağıdaki fonksiyonu kullanır:

Burada N hash tablosunun kapasitesidir. Örneğin, (3,A) key-value çifti, hash fonksiyonu olan ve N=13 kapasitesine sahip hash tablosuna yerleştirilecek olursa,ilk olarak key-value çiftlerinin tutulacağı array’in

indeksine bakar eğer dolu değilse (3,A)’yi 7 numaralı indekse yerleştirir. Eğer 7 indeksi dolu ise bir sonraki i değerini kullanarak boş bir index arar. Yani indeksine bakar boş ise (3,A) ‘yı 8 indeksine yerleştirir ve eğer 8 indeksi doluysa bir sonraki i değerlerini boş bir indeks bulana kadar dener. Bu işlem bu şekilde devam eder.

Bu soruda Quadratic Probing kullanan *QuadraticProbeMap*<K,V> sınıfını aşağıdaki açıklamalara uygun bir biçimde implement etmeniz bekleniyor.

* QuadraticProbeMap(int startCap): Constructor. startCap boyutunda bir bucket array oluşturur.
* void put(K key,V value): Parametre olarak verilen (key, value) çiftini hash table’a yerleştirir. Eğer verilen key değeri mevcutsa eski değeri value değeriyle değiştirir.
* V get(K key): HashTable’da bulunan key değerine sahip elemanin value değerini döner. Eğer key değeri yoksa null döner.
* V delete(K key): HashTable’da bulunan key değerini kaldırır. Bu key değerine sahip value değerini döner. Eğer key mevcut değilse null döner.

Hash fonksiyonu olarak kullanın. **Load factor ( değeri 0.4 ‘ü geçtiğinde o an ki kapasitenin 2 katından büyük en küçük asal sayı olacak şekilde HashTable’ı büyütün.** Örneğin load factor belirtilen eşik değerini aştığında kapasite 17 ise yeni kapasite 37 olması gerekir. Başlangıç kapasitesi 17 olmalıdır. Key değerlerinin Integer, Value değerlerinin String olduğunu varsayabilirsiniz.

**Gönderim**: Soruların cevapları için Lab7.txt adında bir dosya hazırlayın. Ardından QuadraticProbeMap.java ve Lab7.txt dosyalarını uzak.etu.edu.tr platformundan gönderebilirsiniz.