

Soru-1. Toplam 7 bloktan oluşan bir yığın kütüğün üzerinde sıra ile 2 kere baştan sona okuma yapılacaktır. Bu işlem için toplam 3 boş çerçeve kullanılacaksa, yer değiştirme algoritmasının LRU ve MRU olması halinde gerçekleşecek disk erişim sayısını ayrı ayrı hesaplayınız.

Yöntem	İşlem	Disk erişim sayısı
LRU	LRU (ilk okuma) = >1,2,3 / 4,2,3 / 4,5,3/ 4,5,6 / 7,5,6 (7 okuma). İkincisi de bellekte kalan sayfalara bakıldığında 7 kere kullanılacaktır. Toplam 14 okuma olur.	14
MRU	<p>*MRU (ilk okuma) = >1,2,3 / 1,2,4 / 1,2,5/ 1,2,6 / 1,2,7 (7 okuma)/ 1,2,3/1,2,4. İkincisi de bellekte kalan sayfalara bakıldığında 5 kere kullanılacaktır. Toplam 12 okuma olur. (Belleğe çekme zamanına göre MRU)</p> <p>**MRU (ilk okuma) = >1,2,3 / 1,2,4 / 1,2,5/ 1,2,6 / 1,2,7 (7 okuma)/1,3,7/ 1,4,7,. İkincisi de bellekte kalan sayfalara bakıldığında 5 kere kullanılacaktır. Toplam 12 okuma olur. (Son okuma işlem zamanına göre MRU)</p>	11** ve 12*

Soru 2. Her biri 150byte olan 600.000 kayıt bir yığın (heap) kütük içinde saklanmıştır. Kayıtlarda dizinleme için kullanılan biricik anahtarın boyu 22byte'tır. Bu kütüğün saklandığı diskin fiziksel özellikleri yandaki tabloda verilmiştir. Kütüğün saklandığı sistemde en fazla 10 adet ana bellek sayfası mevcuttur ve gösterge boyu 8 byte'tır.

Disk özelliği:	Değeri
Sektör boyu:	512 byte/sektör
İzdeki sektör sayısı:	20 sektör /iz
Öbekteki sektör sayısı:	5 sektör/öbek
Yüzeydeki iz sayısı:	3000 iz / yüzey
Diskteki yüzey sayısı	10 adet ÇİFT taraflı yüzey/disk
Dönüş hızı:	6000 rpm
Yatayda arama zamanı:	8ms

Bu verilere göre aşağıdaki soruları cevaplayınız?

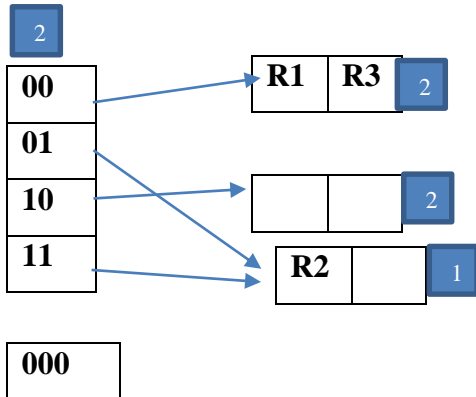
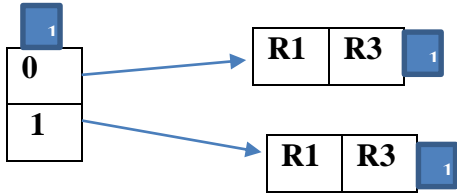
No	Soru	İşlem	Sonuç
1	Yığın kütüğün bloklama faktörü nedir	$512/150 = 3 * 5 = 15$	15
2	Yığın kütüğü saklamak için gereken toplam blok sayısı nedir	$600.000/15 = 40000$	40.000
3	Disk in saklama kapasitesi toplam kaç bloktur	$20/5 = 4$ öbek/iz $4*20 = 80$ öbek/silindir $80*3000 = 240.000$ öbek/disk	240.000
4	Silindir öncelikli yerleşimde yığın kütüğü saklamak için gerekli silindir sayısı kaçtır	$40.000/80 = 500$	500
5	Kütükten tek bir blok okumak için gereken süre nedir	$60.000/6.000 = 10$ msn $8\text{ms} + (10/2 =) 5\text{ms} + (10/4 =) 2.5\text{msn}$ $= 15.5\text{msn}$	15.5 sn
6	Oluşturulacak yoğun dizinin bloklama faktörü kaçtır	$512/(22+8) = 17$ $17 * 5 = 85$	85
7	Yoğun dizinin ikincil bellekte tüketeceği blok sayısı kaçtır	$600.000/85 =$	7059
8	Yoğun dizin üzerine kurulan kaçinci seyrek dizin bellekteki boş sayfalara sığacak büyüklükte olacaktır?	$7059/85 = 84$ blok (1.seyrek) $84/85 = 1$ (2.seyrek)	2. düzey
9	8. soruya göre kütükten bir kayıdı okumak için geçen toplam süre ne olacaktır?	#Disk erişimi= 1.seyrek + yoğun + kütük = 3 blok okuma süresi = 3 *15.5 = 46.5 msn	46.5msn
10	6. sorudaki yoğun dizin üzerine başka bir dizin <u>oluşturulmaz ise</u> , kütükten bir kayıdı okumak için geçen en çok süre ne olur	#Disk erişimi = yoğun dizinde ikili arama + kütük = $(\log_2 7059) + 1 = 14$ ise okuma süresi $14*15,5 = 217$ msn	217msn

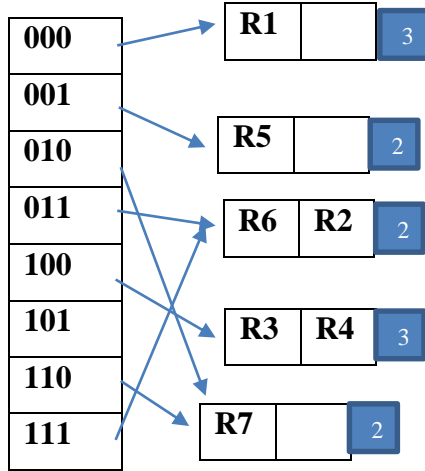
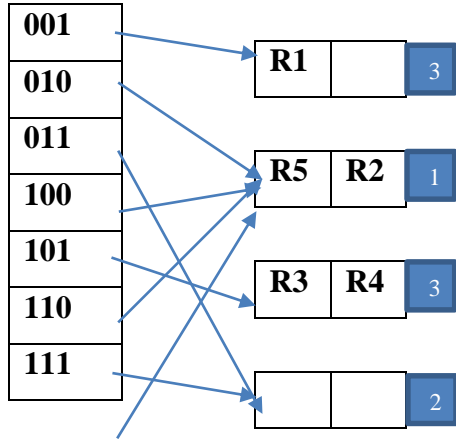
Soru-3. Genişleyebilir anahtarlama yöntemi kullanılarak aşağıda verilen anahtar değerleriyle dizin oluşturulacaktır. Verileri saklamak için kullanılan kapların (bucket) her biri toplam 2 kayıt saklayabilmektedir. Anahtarlama fonksiyonu olarak $H(k) = k \bmod 29$ kullanılmaktadır ve $H(k)$ değerinin en önemsiz bitinden başlayarak anahtarlama yapılmaktadır.

(a) Aşağıda verilen anahtar değerleri için tabloyu doldurunuz.

Kayıt	k	H(k)	Binary
R1	53	24	11000
R2	32	3	00011
R3	70	12	01100
R4	107	20	10100
R5	30	1	00001
R6	65	7	00111
R7	60	2	00010

(b) Yukarıdaki tabloya göre, kayıtlar verildiği sıra ile işlendiğinde oluşacak genişleyebilir anahtarlama yapısını gösteriniz. Her bir kap bölünme işlemini net bir şekilde çiziniz. Rehberin (directory) hangi kapları (bucket) gösterdiğini, yerel derinliği ve genel derinliği belirtiniz.





Soru-4. Bir veritabanı verileri belli bir k alanına (field) göre dizinlemek için genişleyebilir anahtarlama (extendible hashing) yöntemini kullanmaktadır. Söz konusu k alanı 4 byte ile gösterilen rasyonel sayılardır ve değerleri tekrar etmemektedir. Toplam 262.144 kayıt olduğu bilinmektedir. Kap içerisinde anahtar değerine karşılık gelen kaydın yerini gösteren 8 byte boyutunda disk adresi tutulmaktadır. Veritabanı sayfa boyutu (page size) olarak 6 KB kullanılmaktadır. Bu bilgilere göre aşağıdaki soruları cevaplayınız.

(a) Bu veriyi dizinlemek için genel derinliğin alabileceği en düşük değeri hesaplayınız.

6 KB / 12 = 512 entries per bucket

262.144/512 = 512 buckets needed with uniform distribution

$2^d=512 \rightarrow d = 9$

(b) Aşağıdaki 3 soruyu aşağıdaki değişkenler ve yukarıda verilen değerlere göre cevaplayınız.

Cevabınızı mümkün olduğunca sadeleştiriniz.

N : Kullanılan toplam kap sayısı (number of buckets)

R : Bir sayfanın okunma zamanı

D : Directory boyutu (byte olarak)

- (i) Ara bellekte sadece directory bulunuyorsa, ' k değeri 20,2 olan kayıdı getir' gibi bir sorguyu işleme zamanını belirtiniz.

1 bucket okumak için +1 kütük okuma = 2 erişim

- (ii) Dizin için ara bellekten faydalanılmıyorsa, $k=20,2$ gibi bir sorguyu işleme zamanını yazınız.

*$(\text{ceil}(D/6K) + 1\text{bucket} + 1\text{ kütük}) * R$*

- (iii) Rasyonel sayı olan k anahtarı için ' $k > 10$ & $k < 20$ koşullarını sağlayan kayıtların getirilmesi' gibi bir aralık sorgusu verilirse kayıtların bulunma zamanını belirtiniz (Ara bellekte sadece directory bulunuyorsa).

*Read all the buckets. $N * R + M * R$*

Soru-5. Hangi işlem senaryosunda yoğun dizinle birlikte kullanılan sıralı kütük, seyrek dizinle birlikte kullanılan sıralı kütüğe göre daha avantajlı bir kullanım sunar (ipucu: arama zamanındaki disk erişimini azaltır) (10 puan)

Eğer aranan kayıt veri dosyasında bulunmuyorsa/min-max tipi sorgular/count tipi sparse index'de bucket okumak gerekir. Fakat yoğun dizinde sadece dizine bakıp kaydın olmadığını bilebiliriz.

Puan alan diğer cevaplar:

- *Yoğun dizinde her kaydın bucket içinde nerede olduğunu biliyoruz, bu yüzden daha hızlı olur. Disk erişimi sayısı etkilenmiyor. 4 puan*
- *Eğer tekrar eden değerler varsa: Bu durumda eğer Seyrek dizinde derste bahsettiğimiz gibi bir çözüm uygulanmazsa önündeki ve sonrasındaki bucketlerin de okunması gerekebilir. 8 puan.*