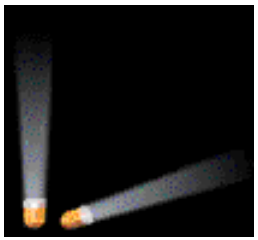
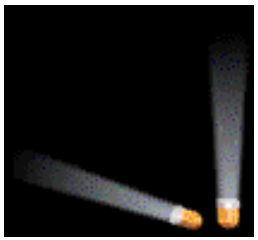
F.CS203 - Өгөгдлийн бүтэц ба алгоритм 2022-2023

Д.Золбоо

МТ-н салбарын багш

Толь бичиг

- Хосуудын цуглуулга.
  - (key, element)
  - Хос бүр өөр түлхүүртэй.
- Үйлдлүүд.
  - get(theKey)
  - put(theKey, theElement)
  - remove(theKey)

Хэрэглээ

- F.CS203 –г сонгосон оюутнууд.
  - (key, element) = (оюутны нэр, бие даалт болон шалгалтын дүнгийн шугаман жагсаалт)
  - Бүх түлхүүр ялгаатай.
- Ө.Дөлгөөн гэсэн түлхүүртэй элементийн авах
  - get()
- Д.Туяа гэсэн түлхүүртэй элементийг өөрчлөх
  - put().
  - remove().

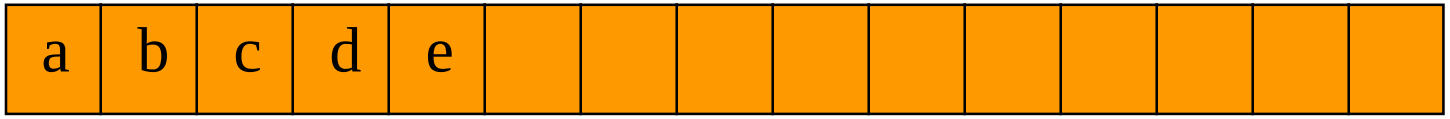
Давхцалттай толь бичиг

- Түлхүүр давхцаж болно.
- Үгсийн толь бичиг.
  - Хос нь (үг, утга).
  - Нэг үг хэд хэдэн утгатай байж болно.
    - (гар, хүний эрхтэн)
    - (гар, гадагшлах хөдөлгөөн)
    - (гар, компьютерийн оруулах төхөөрмж)
    - гэх мэт.

Шугаман жагсаалтаар дүрслэх

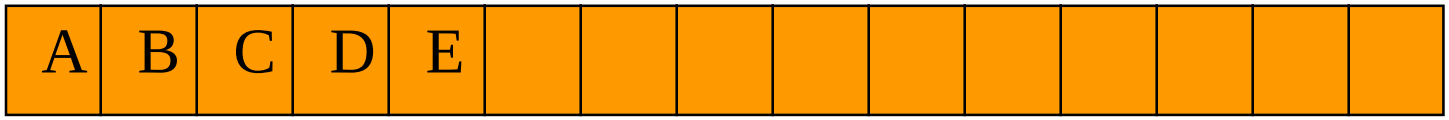
- $L = (e_0, e_1, e_2, e_3, \ldots, e_{n-1})$
- e_i бүхэн (key, element).
- 5-хостой толь бичиг D = (a, b, c, d, e).
  - a = (aKey, aElement), b = (bKey, bElement), г.м.
- Массив эсвэл Холбоост дүрслэл.

Массив дүрслэл



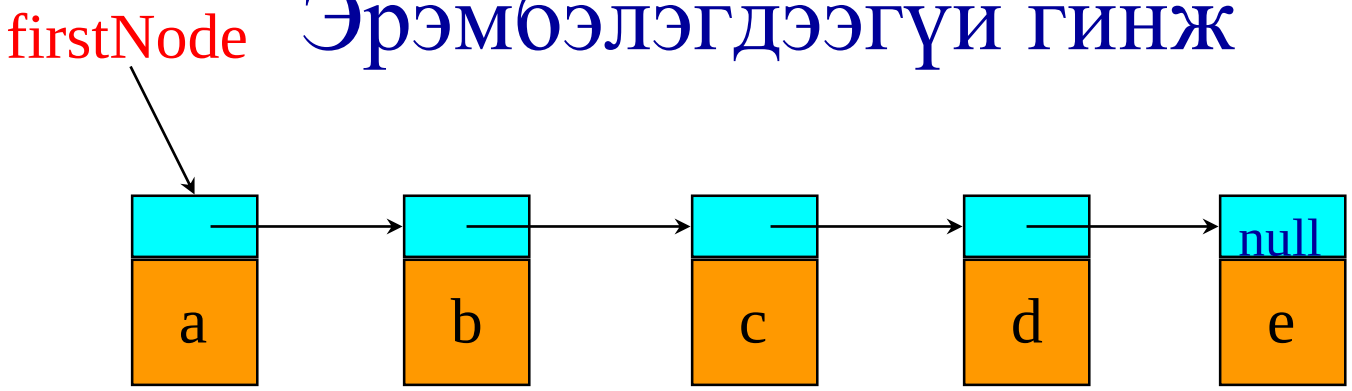
- get(theKey)
- put(theKey, theElement)
- remove(theKey)

Эрэмбэлэгдсэн массив



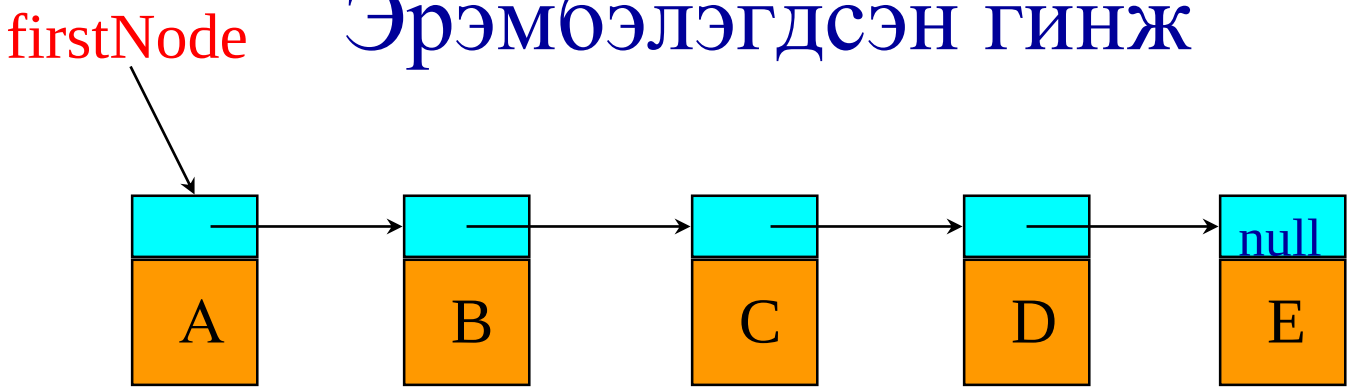
- Элементүүд түлхүүрийн өсөх дарааллаар байрлана.
- get(theKey)
  - O(log size)
- put(theKey, theElement)
  - O(log size) давхцлыг шалгахад, O(size) нэмэхэд
- remove(theKey)
  - O(size)

Эрэмбэлэгдээгүй гинж



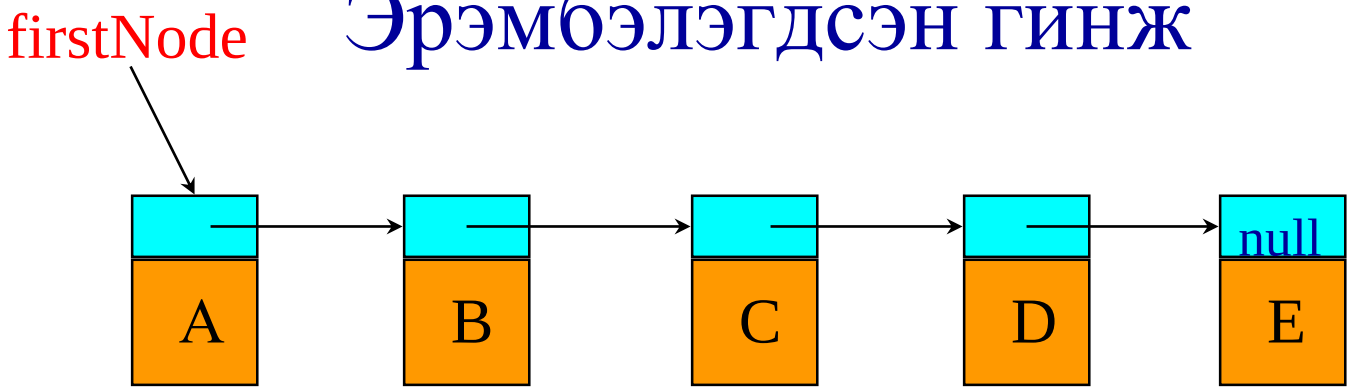
- get(theKey)
  - O(size)
- put(theKey, theElement)
  - O(size) давхцлыг шалгахад, O(1) зүүн талд нь нэмэхэд
- remove(theKey)
  - O(size)

Эрэмбэлэгдсэн гинж



- Элементүүд түлхүүрийн өсөх дараалалтай.
- get(theKey)
  - O(size)
- put(theKey, theElement)
  - O(size) давхцлыг шалгахад, O(1) зөв газар нь нэмэхэд

Эрэмбэлэгдсэн гинж



- Элементүүд түлхүүрийн өсөх дараалалтай
- remove(theKey)
  - O(size)
-

Алгасах жагсаалт

- Муу тохиолдолд get, put, remove - O(size).
- Дундаж хугацаа – O(log size)
- Skip list – алгасах жагсаалтыг бид алгасна

Хэш хүснэгт

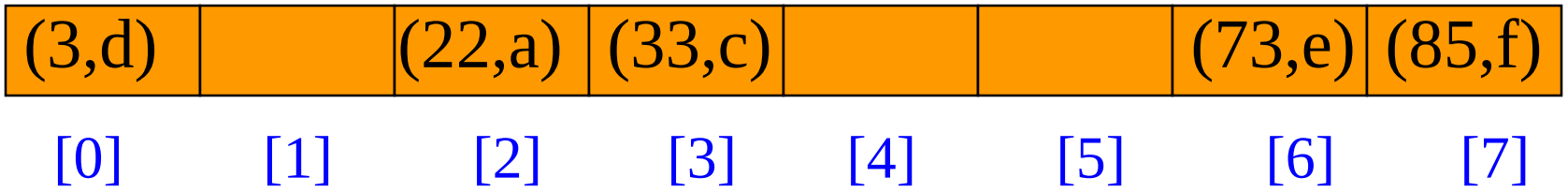
- Муу тохиолдолд get, put, remove - O(size).
- Хүссэн хугацаа O(1).

Хүслийн Хэш

- 1D массив(хүснэгт) ашиглая table[0:b-1].
  - Массивын байршил бүр багц.
  - Ер нь багц толь бичгийн зөвхөн нэг хосыг хадгалах ёстой.
- Ашиглах хэш функц f түлхүүр үг k хүснэгтийн индекст хувиргах ёстой [0, b-1].
  - f(k) бол түлхүүр k -ийн багцны үүр
- Толь бичгийн (key, element) хос бүр table[f[key]] гэсэн багцны үүрэнд хадгалагдана

Хүслийн хэшийн жишээ

- Хосууд: (22,a), (33,c), (3,d), (73,e), (85,f).
- Хэш хүснэгт table[0:7], b = 8.
- Хэш функц key/11.
- Хосууд хүснэгтэд дараах байдлаар хадгалагдана:



- get, put, remove - ажиллах хугацаа O(1).

Явал буруу тийш явах вэ?

(3,d)		(22,a)	(33,c)			(73,e)	(85,f)
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]

- (26,g) хаашаа явах вэ?
- Нэг багцны үүртэй түлхүүрүүдийг синоним гэнэ
  - 22 ба 26 бол ашиглаж байгаа хэш функцийн хувьд синонимууд.
- (26,g) –д харгалзах багцны үүр эзлэгдсэн байна.

Явал буруу тийш явах вэ?

(3,d)		(22,a)	(33,c)			(73,e)	(85,f)

- Өөр түлхүүртэй шинэ хосод харгалзах багцны үүр эзлэгдсэнээс collision-зөрчил үүснэ.
- Шинэ хосод харгалзах багцны үүрэнд зай байхгүйгээс overflow-халилт үүснэ.
- Багцны үүр зөвхөн нэг хосыг хадгалдаг бол зөрчил болон халилт зэрэг үүснэ.
- Халилтыг шийдэх арга хэрэгтэй.

Хэш хүснэгтийн асуудлууд

- Хэш функцийг сонгох.
- Халилтыг зохицуулах арга.
- Хэш хүснэгтийн хэмжээ (Багцны тоо).

Хэш функцүүд

- Хоёр хэсэг:
  - Түлхүүр бүхэл биш бол бүхэл болгох.
    - Шийдэх арга нь hashCode().
  - Бүхэл тоог багцны үүрт буулгах.
    - f(k) функц [0, b-1] мужид бүхэл утгатай, үүнд b бол хүснэгт дэх багцны тоо.

Тэмдэгт мөрийг бүхэл тоо руу хувиргах

- Java –ийн тэмдэгт бүр 2 байт урттай.
- int бол 4 байт.
- 2 тэмдэгтийн s хэлхээсийг давтагдашгүй 4 байт int –д хувиргахдаа:

```
int answer = s.charAt(0);
answer = (answer << 16) + s.charAt(1);
```

- 2 тэмдэгтээс урт хэлхээст давтагдашгүй int дүрслэл байхгүй.

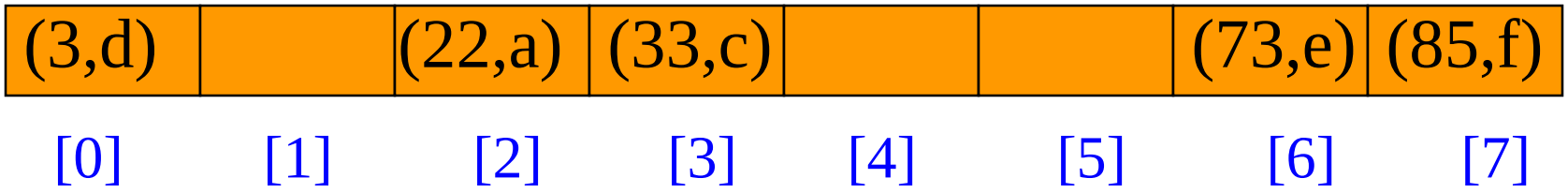
Тэмдэгт мөрийг бүхэл тоо руу хувиргах

```
public static int integer(String s)
{
int length = s.length();
        //s-ийн тэмдэгтийн тоо
int answer = 0;
if(length % 2 == 1)
{//урт нь сондгой бол
        answer = s.charAt(length-1);
        length--;
}
```

Тэмдэгт мөрийг сөрөг биш бүхэл тоо руу хувиргах

```
//урт нь тэгш бол
For (int i = 0; i<length; i+=2)
{//нэг удаа 2 тэмдэгтийг
        answer+=s.charAt(i);
        answer+=((int)s.charAt(i+1))<<16;
}
return (answer < 0) ? –answer : answer;
}
```

Багцны үүрийн буулгалт



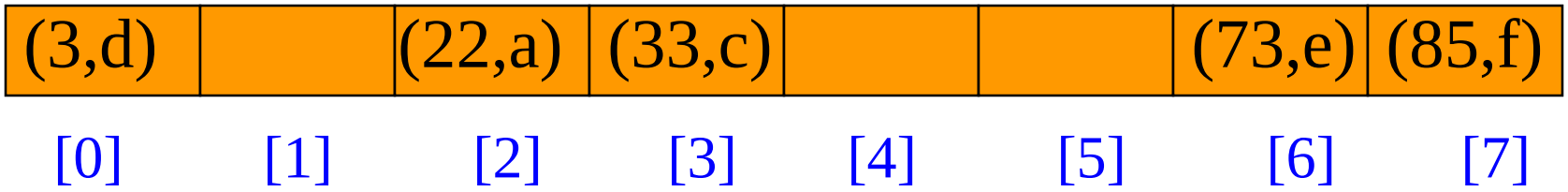
- Хамгийн нийтлэг арга бол хуваалт(divisor).

homeBucket =

Math.abs(theKey.hashCode()) % divisor;

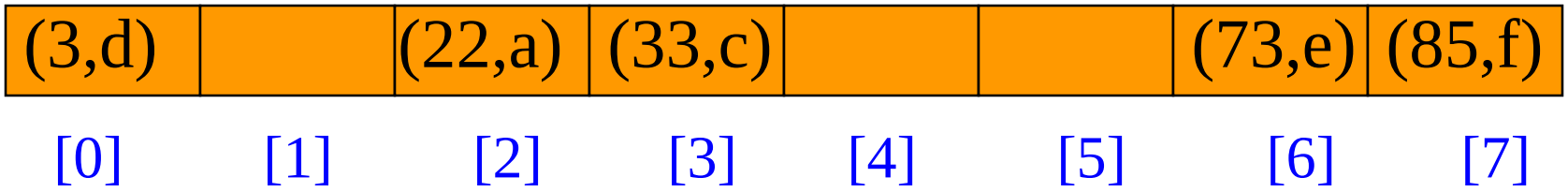
- divisor багцны тоо b -тэй тэнцүү
- 0 <= homeBucket < divisor = b

Жигд Хэш Функц



- keySpace бол байж болох түлхүүрийн олонлог болог.
- Хэш функц keySpace олонлогийн түлхүүрийг багцад тусгахдаа дунджаар ижил тооны түлхүүр нэг багцад тусч байвал функцийг жигд хэш функц гэдэг.

Жигд Хэш Функц



- Өөрөөр хэлбэл, санамсаргүй сонгосон түлхүүр багц i-д тусах магадлал 1/b, 0<=i<b
- Жигд хэш функц түлхүүр санамсаргүй сонгогдох тохиолдолд халилтыг минимум болгодог.

Хуваалтын хэш

- keySpace = бүх int-үүд
- Ямар ч b-ийн хувьд дунджаар $2^{32}/b$ тооны int-үүд багц i-д тусдаг (хуваагддаг)
- Иймд, keySpace = бүх int-үүд бол хуваалтын нэгэн жигд хэш функц болно
- Амьдралд түлхүүрүүд хамааралтай байдаг
- Тэгэхээр, хуваагч b-ийн сонголт багцын үүрийн буулгалтанд нөлөөлдөг

Хуваагчийг сонгох

- Түлхүүрүүд ер нь хамааралтай байдгаас тэгш, сондгой багцны үүрт буулгалт давамгайлах тал байдаг.
- Хуваагч тэгш тоо бол, сондгой бүхэл тоотой багцад, тэгш бүхэл тэгш багцад тусдаг.
  - 20%14 = 6, 30%14 = 2, 8%14 = 8
  - 15%14 = 1, 3%14 = 3, 23%14 = 9

Хуваагчийг сонгох

- Хуваагч сондгой тоо бол, сондгой (тэгш) бүхэл дурын үүрд хуваагдаж болдог.
  - 20%15 = 5, 30%15 = 0, 8%15 = 8
  - 15%15 = 0, 3%15 = 3, 23%15 = 8
- Багцны үүрт нэгэн жигд тараах илүү боломжтой.
- Иймд тэгш хуваагчийг битгий ашигла.

Хуваагчийг сонгох

- Амьдралд жигд биш тархалт хуваагчийг 3,5,7,... мэтийн анхны тооны үржвэр байдлаар сонгосноос болдог
- Хэрвээ p нь b–ийн хувьсагч бол p өсөх тутам энэ нөлөөлөл багасдаг
- Зөв сонголтоор b анхны тоо байх нь чухал
- Эсвэл, b-г сонгохдоо 20–оос доош тоонд хуваагддагүй байх явдал

Java.util.HashTable

- Сондгой тоог хуваагч болгон ашигладаг.
- Ингэснээр олон хосыг толь бичигт оруулах зорилгоор хэш хүснэгтийн хэмжээг өөрчлөх боломж олгодог.
  - Жишээ нь, массивыг 2 дахин ихэсгэхэд, b (сондгой) урттай 1D массив table –ийн уртыг 2b+1 болгоно(мөн сондгой).

Халилтыг зохицуулах

- Шинэ хос (key, element) -ийн хувьд багцны үүр дүүрэн бол халилт үүсдэг.
- Халилтыг зохицуулахдаа:
  - Хэш хүснэгтээс голдуу дүүрэн бус байдаг багцыг хай.
    - Шугаман тандалт.
    - Квадрат тандалт.
    - Санамсаргүй тандалт.
  - Багц бүрт ижил үүртэй бүх хосуудын жагсаалтыг хадгалах замаар халилтаас зайлсхийж болно.
    - Массив шугаман жагсаалт.
    - Гинж.