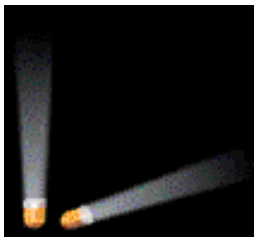
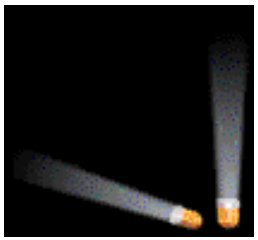
Толь бичиг





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

Хэрэглээ

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

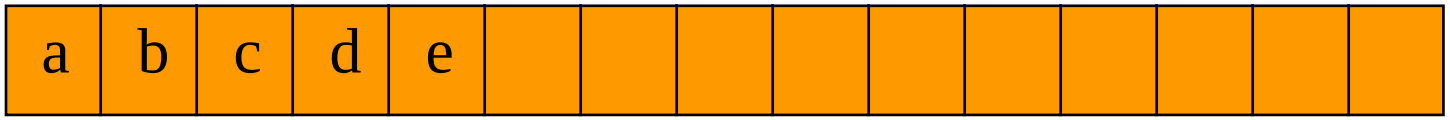
Давхцалттай толь бичиг

- 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

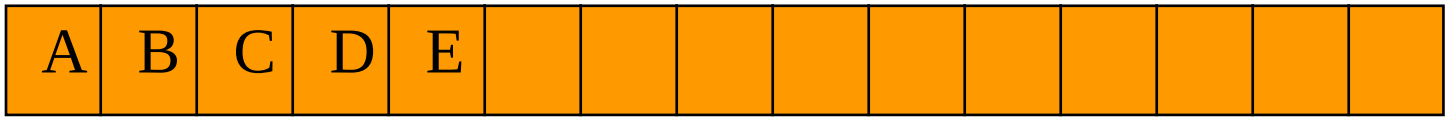
- $L = (e_0, e_1, e_2, e_3, \ldots, e_{n-1})$
- e_i бүхэн (key, element).
- 5-хостой толь бичиг D = (a, b, c, d, e).
  - a = (aKey, aElement), b = (bKey, bElement), г.м.
- Массив эсхүл Холбоост дүрслэл.

Массив дүрслэл



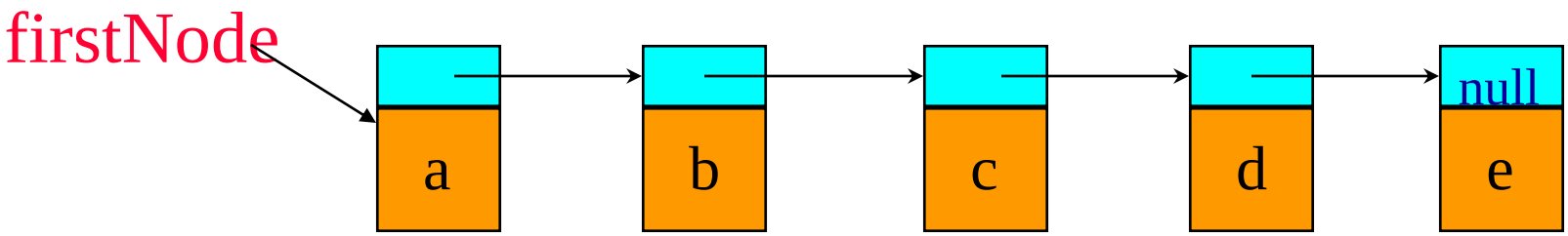
- get(theKey)
- put(theKey, theElement)
- remove(theKey)

Эрэмбэлэгдсэн массив



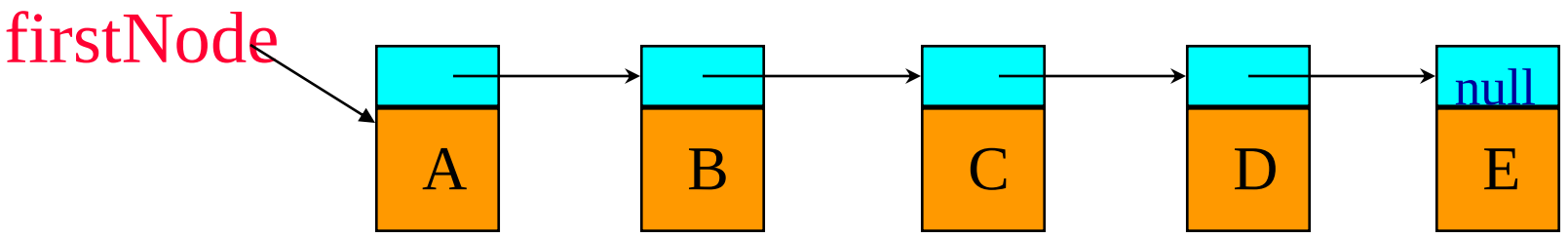
- элементүүд түлхүүрийн өсөх дарааллаар байрлана.
- get(theKey)
  - O(log size)
- put(theKey, theElement)
  - O(log size) давхцлыг шалгахад, O(size) нэмэхэд.
- remove(theKey)
  - O(size).

Эрэмбэлэгдээгүй гинж



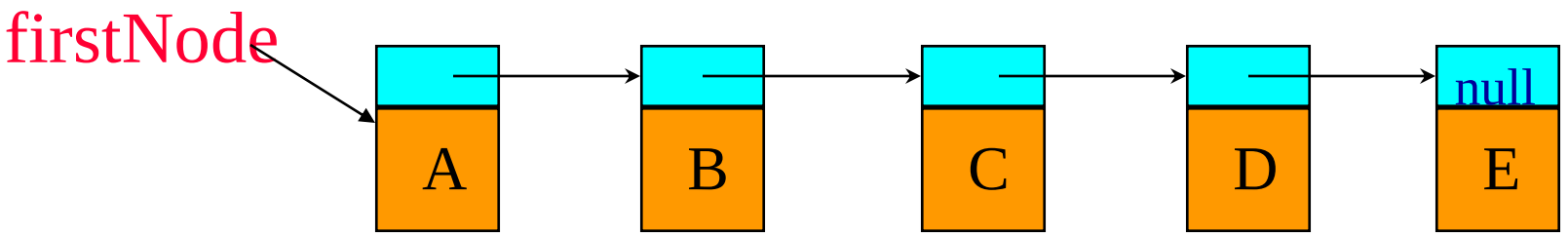
- get(theKey)
  - O(size)
- put(theKey, theElement)
  - O(size) давхцлыг шалгахад, O(1) зүүн талд нь нэмэхэд.
- remove(theKey)
  - O(size).

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 lists – алгасах жагсаалтыг бид алгасна

Хэш хүснэгт

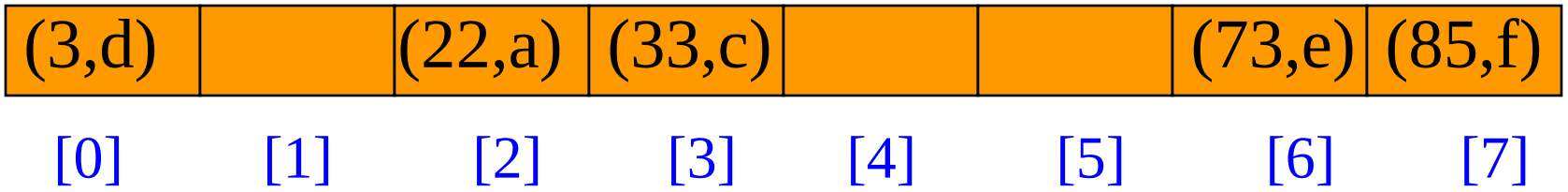
- Муу тохиолдолд get, put, remove - O(size).
- Дундаж хугацаа O(1).

Төгс 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 нь түлхүүр k – г [0, b-1] завсар дахь хүснэгтийн индекст хувиргах ёстой.
  - f(k) бол түлхүүр k -ийн багцны үүр
- Толь бичгийн (key, element) хос бүр table[f[key]] гэсэн багцны үүрэнд хадгалагдана

Төгс 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 тэмдэгтэй s тэмдэгт мөрийг давтагдашгүй 4 байт int –д хувиргахдаа:

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

- 2 тэмдэгтээс урт тэмдэгт мөрөнд давтагдашгүй int дүрслэл байхгүй.

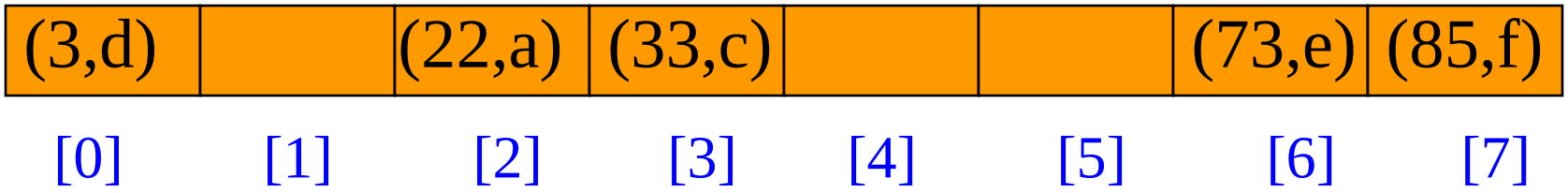
Тэмдэгт мөрийг сөрөг биш бүхэл тоо руу хувиргах

```
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



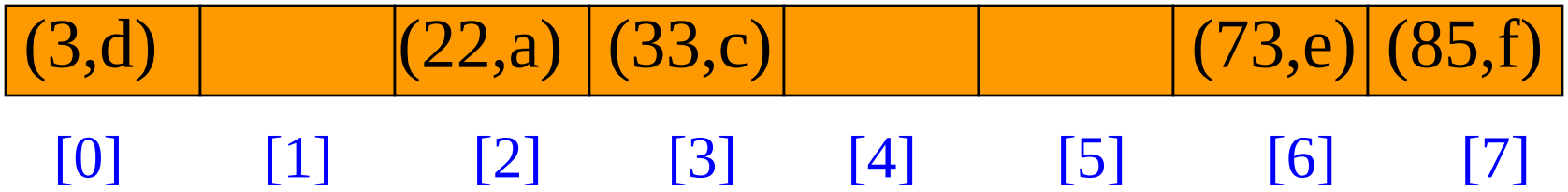
- Хамгийн нийтлэг арга бол хуваалт(divisor).

homeBucket =

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

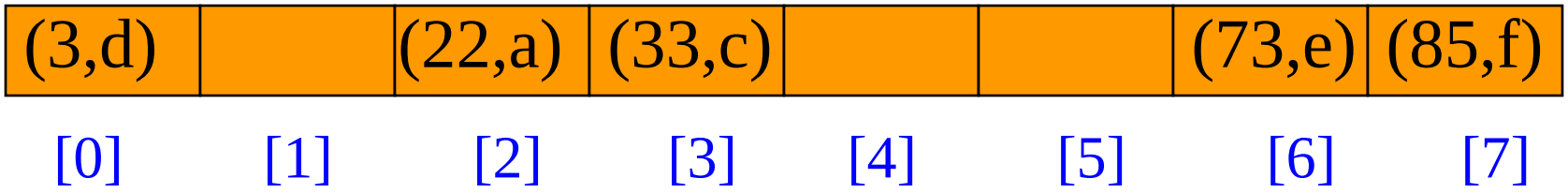
- divisor багцны тоо b -тэй тэнцүү
- 0 <= homeBucket < divisor = b

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 арга 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 сондгой 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 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 хуваагч бол p өсөх тутам энэ нөлөөлөл багасдаг.
- Зөв сонголтоор b анхны тоо байх нь чухал.
- Эсхүл, b-г сонгохдоо 20 -оос доош тоонд хуваагддаггүй байх явдал

Java.util.HashTable

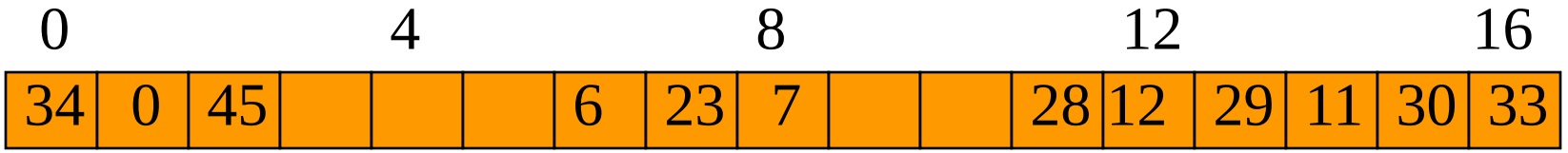
- Сондгой тоог хуваагч болгон ашигладаг.
- Ингэснээр олон хосыг толь бичигт оруулах зорилгоор хэш хүснэгтийн хэмжээг өөрчлөх боломж олгодог.
  - Жишээ нь, массивыг 2 дахин ихэсгэхэд, b (сондгой) урттай 1D массив table –ийн уртыг 2b+1 болгоно(мөн сондгой).

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.

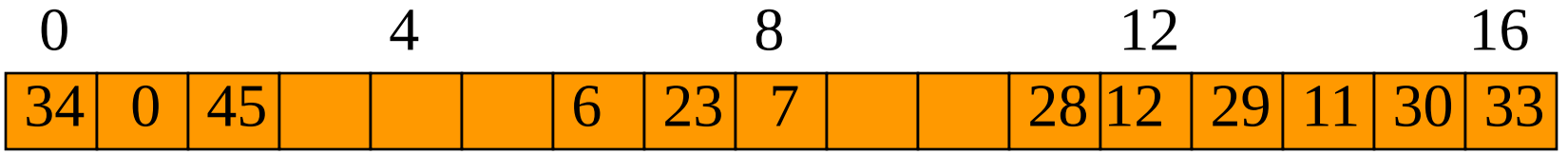
Шугаман тандалт – Get ба Put

- divisor = b (багцын тоо) = 17.
- Багцын үүр = key % 17.

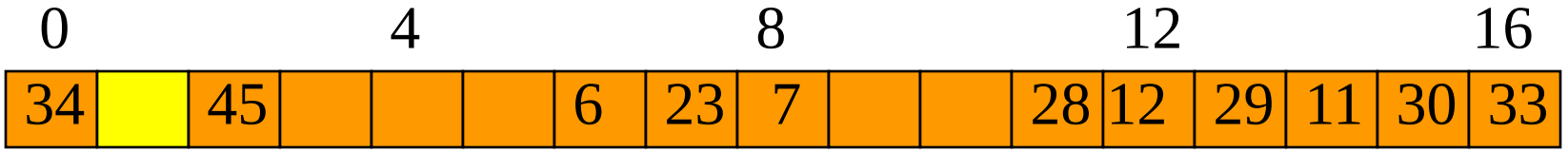


- 6, 12, 34, 29, 28, 11, 23, 7, 0, 33, 30, 45 гэсэн түлхүүртэй хосуудыг хийлээ

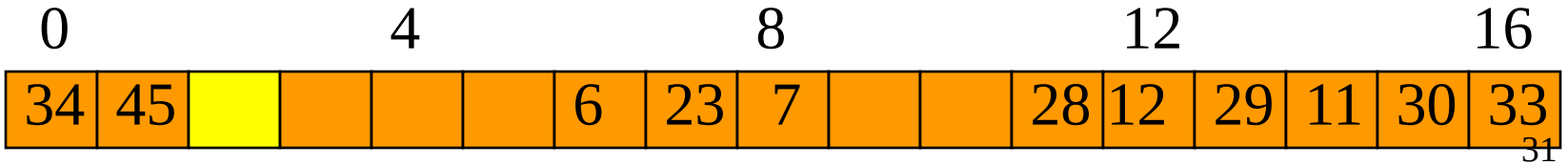
Шугаман тандалт – Remove



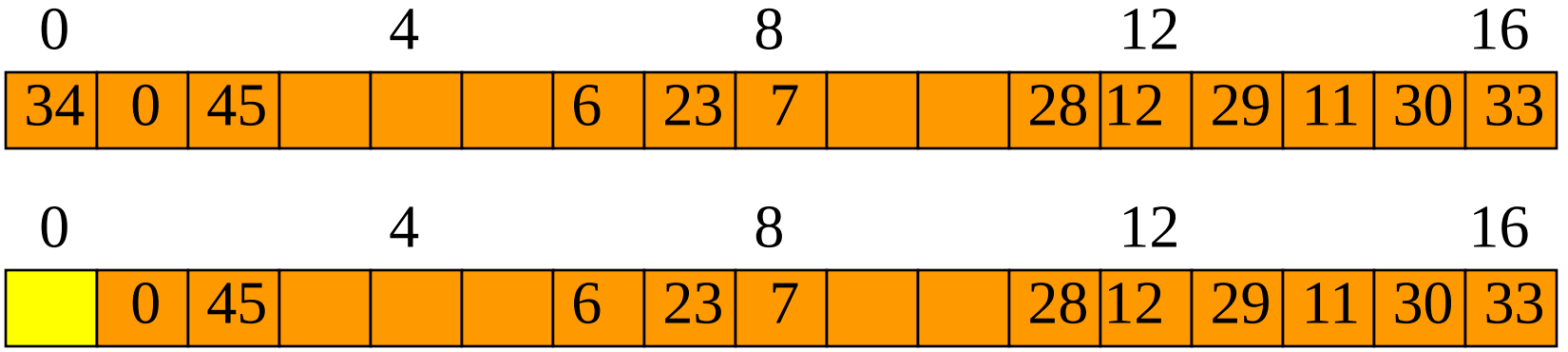
- remove(0)



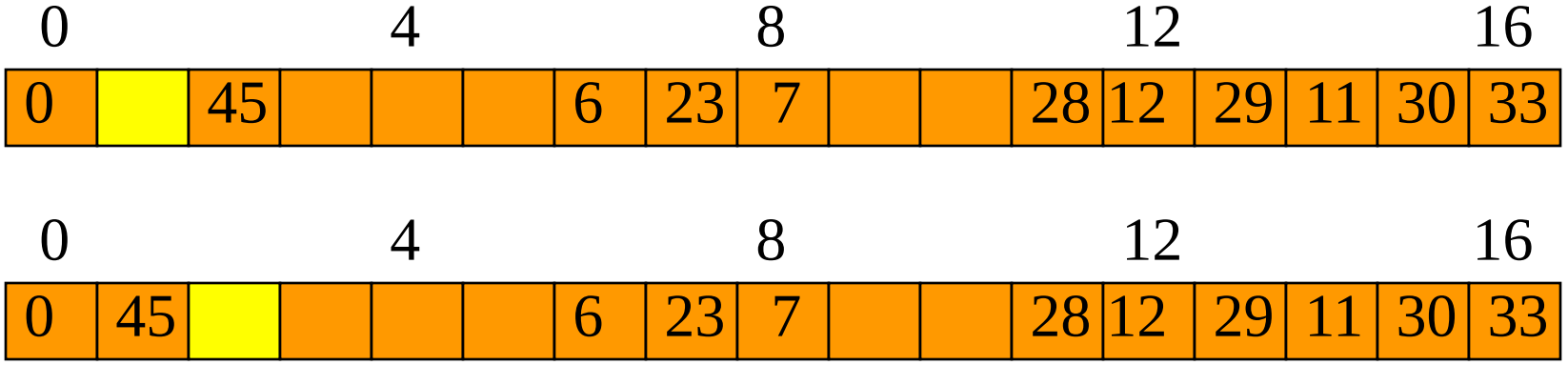
- Чөлөөлөгдсөн багцыг ашиглаж болох хосыг хайх.



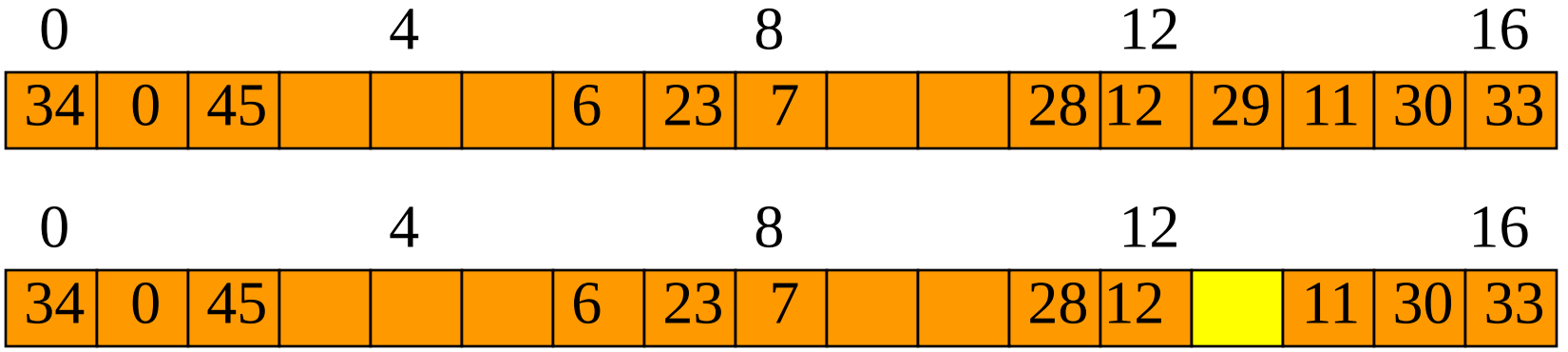
Шугаман тандалт – remove(34)



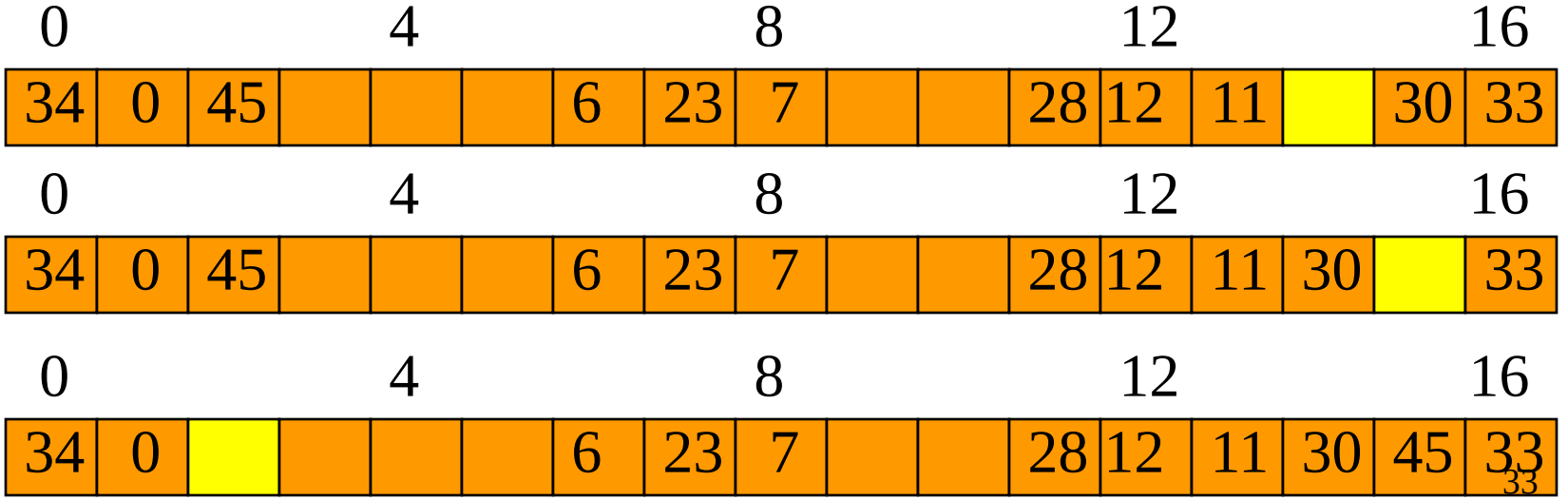
- Чөлөөлөгдсөн багцыг ашиглаж болох хосыг хайх.



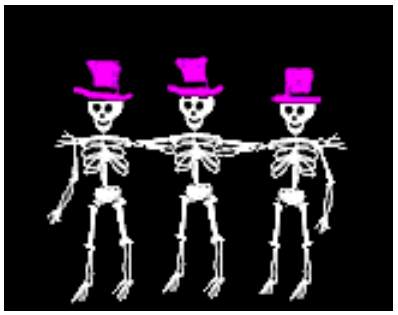
Шугаман тандалт – remove(29)



- Чөлөөлөгдсөн багцыг ашиглаж болох хосыг хайх.



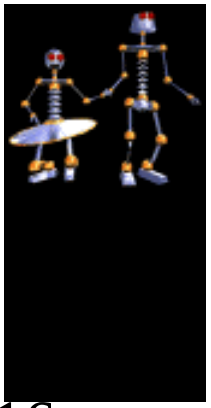
Шугаман тандалтын үзүүлэлт

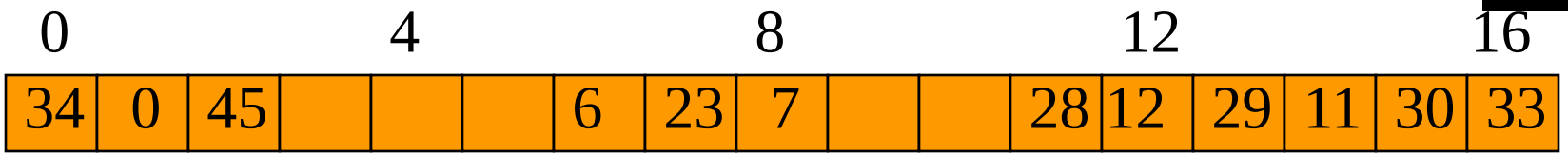


0			4				8				12					16
34	0	45				6	23	7			28	12	29	11	30	33

- get/put/remove үйлдлийн муу тохиолдлын хугацаа $\Theta(n)$, үүнд n – хүснэгт дэх хосын тоо.
- Бүх хос нэг үүрт ороход үүснэ.

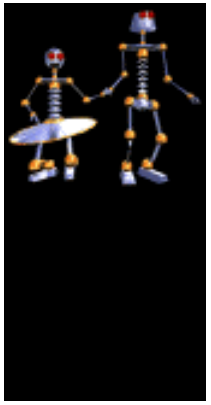
Дундаж үзүүлэлт





- α = ачааллын нягтрал = (хосын тоо)/b.
  - α = 12/17.
- S_n = амжилттай хайлтаар шалгах багцын (дундаж) тоо (n - том тоо бол)
- U_n = амжилтгүй хайлтаар шалгах багцын (дундаж) тоо (n - том тоо бол)
- тэгвэл put , remove үйлдлүүдийн хугацаа U_n -аар тодорхойлогдоно

Дундаж үзүүлэлт



- $S_n \sim ½(1 + 1/(1 - \alpha))$
- $U_n \sim ½(1 + 1/(1 - \alpha)^2)$
- $0 <= \alpha <= 1$.

α	S_n	U_n
0.50	1.5	2.5
0.75	2.5	8.5
0.90	5.5	50.5

$\alpha <= 0.75$ байхыг зөвлөдөг.

Хэш хүснэгтийн зохиомж

- Өгөгдсөн шаардлагаас хамаарч, ачааллын нягтралын зөвшөөрөгдөх дээд хэмжээг тогтоох.
- Амжилттай хайлт хийхэд 10 –с илүүгүй харьцуулалт хэрэгтэй бол (шаардлага).
  - $S_n \sim ½(1 + 1/(1 - \alpha))$
  - $\alpha <= 18/19$
- Амжилтгүй хайлт хийхэд 10 –с илүүгүй харьцуулалт хэрэгтэй бол (шаардлага).
  - $U_n \sim ½(1 + 1/(1 - \alpha)^2)$
  - $\alpha <= 4/5$
- Иймд $\alpha <= \min\{18/19, 4/5\} = 4/5$.

Хэш хүснэгтийн зохиомж

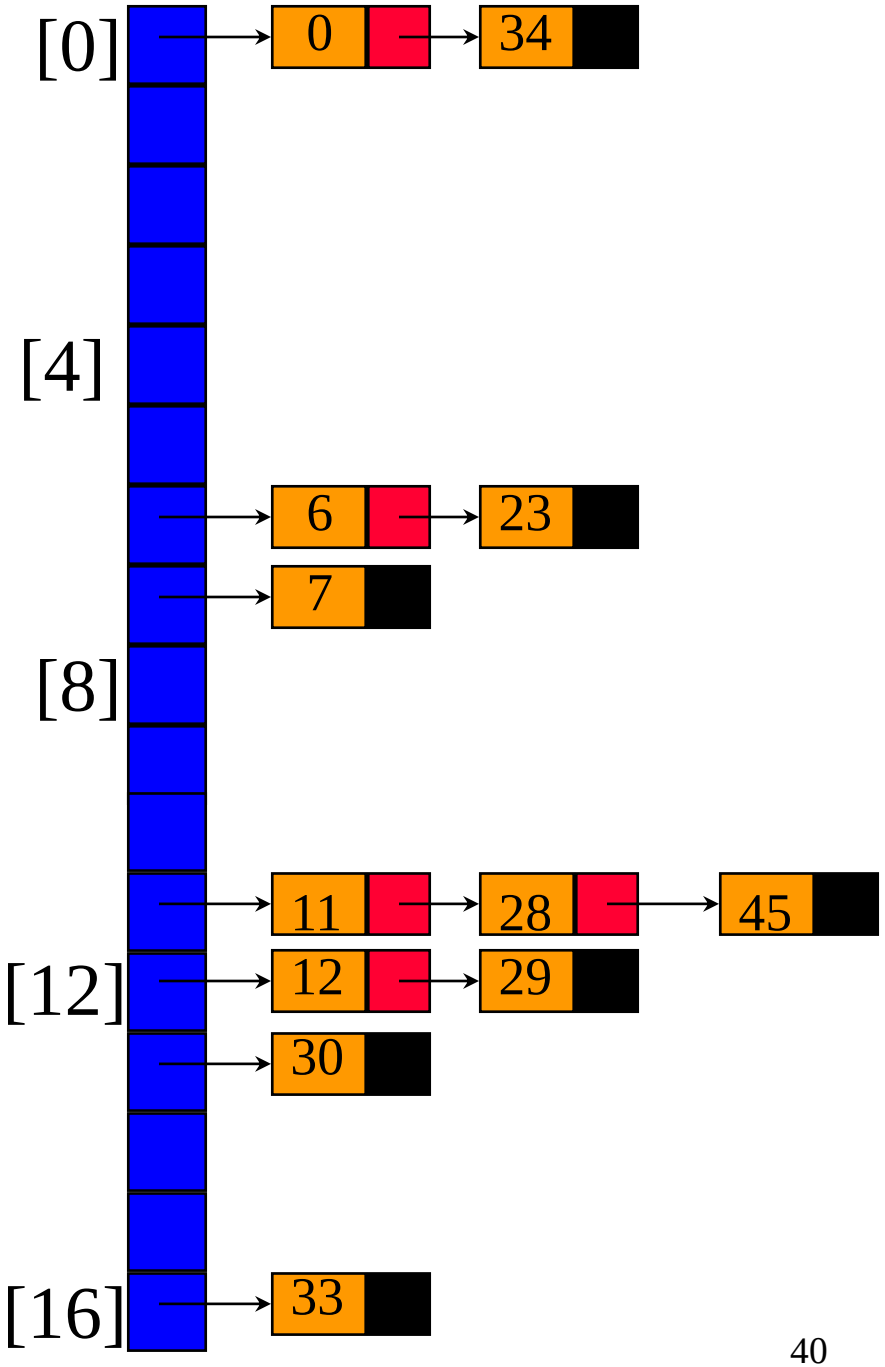
- Динамик хүснэгтийн хэмжээ.
  - Ачааллын нягтрал хүссэн хэмжээнээс хэтэрвэл (4/5 манай жишээнд), хэш хүснэгтийн хэмжээг одоогийнхоос нь ойролцоогоор 2 дахин нэмэгдүүлнэ.
- Тогтсон хүснэгтийн хэмжээ.
  - Хосын максимум тоог мэднэ.
  - 1000 -с илүүгүй хостой.
  - Ачааллын нягтрал <= 4/5 => b >= 5/4*1000 = 1250.
  - b (ө.х. divisor) 20 –с доошхи анхны тоонд хуваагддаггүй сондгой тоо, эсхүл анхны тоо байхаар сонго.

Синонимын шугаман жагсаалт

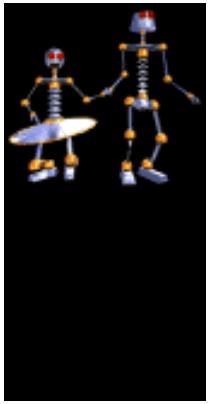
- Багц бүрт ижил үүртэй хосуудын шугаман жагсаалтыг халдгалах.
- Шугаман жагсаалт эрэмбэлэгдсэн байж болно.
- Шугаман жагсаалт нь массив эсхүл гинж байж болно.

Эрэмбэлэгдсэн гинж

- 6, 12, 34, 29, 28, 11, 23, 7, 0, 33, 30, 45 түлхүүртэй хосуудыг хийх
- Багцны үүр = key % 17.



Дундаж үзүүлэлт

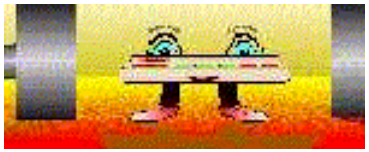


- $\alpha >= 0$ байна.
- Гинжний дундаж урт α.
- $S_n \sim 1 + \alpha /2$.
- $U_n <= \alpha$, $\alpha < 1$ бол
- $U_n \sim 1 + \alpha /2$, $\alpha >= 1$ бол

java.util.Hashtable

- Эрэмбэлээгүй гинж.
- Анхдагч хуваагч b = divisor = 101
- Анхдагч $\alpha <= 0.75$
- Ачааллын нягтрал зөвшөөрөгдсөн максимум нягтралаас давбал хэш хүснэгтийн шинэ урт newB = 2b+1.

Өгөгдлийг шахах



- Өгөгдлийн хэмжээг багасгах.
  - Санах ойг багасгаж, ингэснээр санах ойн зардлыг багасгана.
    - Шахалтын зэрэг = анхны хэмжээ/шахагдсан хэмжээ
  - Өгөгдөл дамжуулах, хүлээн авах хугацааг багасгана.

Хаягдалгүй, хаягдалтай шахалт

- compressedData = compress(originalData)
- decompressedData = decompress(compressedData)
- originalData = decompressedData, бол хаягдалгүй шахалт
- originalData != decompressedData, бол хаягдалтай шахалт

Хаягдалгүй, хаягдалтай шахалт

- Хаягдалтай шахагч, хаягдалгүй шахагчаас илүү өндөр шахалтын зэрэгтэй
  - магадгүй 100 : 2.
- Хаягдалгүй шахалтыг жишээ нь текст файлын хэрэглээнд ашигладаг.
- Хаягдалтай шахалтыг дүрсний хэрэглээнд өргөн хэрэглэдэг.
  - Видео дамжуулахад бага зэргийн хаягдлыг хүний нүд ялгадаггүй.

Текст шахалт

- Хаягдалгүй шахах нь чухал.

•Түгээмэл тархсан zip болон Unix-н compress шахагч LZW(1984) (Lempel-Ziv-Welch) аргыг ашигладаг.

LZW шахалт

- Эх текстэд орсон тэмдэгтийн цувааг динамик байдлаар тодорхойлогддог кодоор сольдог.
- Кодын хүснэгтийг шахагдсан текстэд кодчилодоггүй. Учир нь буцааж задлахад хэрэг болдог.

LZW шахалт

- Текстийн тэмдэгтэд хязгаарлалт хийе {a, b}.
  - Амьдралд цагаан толгой ASCII олонлогийн 256 тэмдэгттэй.
- Цагаан толгойн тэмдэгтүүдэд олгох кодыг 0 -ээс эхлэн дугаарлая
- Анхны кодын хүснэгт:

code	0	1
key	a	b

LZW шахалт

code	0	1
key	a	b

- Эх текст = abababbabaabbabbaabba
- Эх текстийг зүүнээс баруун тийш гүйлгэх замаар шахья.
- Кодын хүснэгтэд код нь орсон хамгийн урт угтвар p –г хайна.
- p –г түүний код pCode –оор дүрсэлж, дараачийн боломжтой кодыг pc -д онооно. Үүнд c бол шахах текстийн дараачийн тэмдэгт.

LZW шахалт

code	0	1	2
key	a	b	ab

- Эх текст = abababbabaabbabbaabba
- p = a
- pCode = 0
- c = b
- a -г 0 -ээр дүрслээд ab –г кодын хүснэгтэд нэмнэ.
- Шахсан текст = 0

LZW шахалт

code	0	1	2	3
key	a	b	ab	ba

- Эх текст = abababbabaabbbabbbaabba
- Шахсан текст = 0
- p = b
- pCode = 1
- c = a
- b -г 1 а-ээр дүслээд ba –г кодын хүснэгтэд нэмнэ.
- Шахсан текст = 01

LZW шахалт

code	0	1	2	3	4
key	a	b	ab	ba	aba

- Эх текст = abababbabaabbbabbaabba
- Шахсан текст = 01
- p = ab
- pCode = 2
- c = a
- ab -г 2 -оор дүслээд aba кодын хүснэгтэд нэмнэ.
- Шахсан текст = 012

LZW шахалт

code	0	1	2	3	4	5
key	a	b	ab	ba	aba	abb

- Эх текст = ababababbabaabbbabbaabba
- Шахсан текст = 012
- p = ab
- pCode = 2
- c = b
- ab -г 2 -оор дүрслээд abb –г кодын хүснэгтэд нэмнэ.
- Шахсан текст = 0122

LZW шахалт

code	0	1	2	3	4	5	6
key	a	b	ab	ba	aba	abb	bab

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 0122
- p = ba
- pCode = 3
- c = b
- ba -г 3 –аар дүрслээд bab –г кодын хүснэгтэд нэмнэ.
- Шахсан текст = 01223

LZW шахалт

code	0	1	2	3	4	5	6	7
key	a	b	ab	ba	aba	abb	bab	baa

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 01223
- p = ba
- pCode = 3
- c = a
- ba -г 3 -аар дүрслээд baa –г кодын хүснэгтэд нэмнэ.
- Шахсан текст = 012233

LZW шахалт

code	0	1	2	3	4	5	6	7	8
key	a	b	ab	ba	aba	abb	bab	baa	abba

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233
- p = abb
- pCode = 5
- c = a
- abb -г 5 -аар дүрслээд abba –г кодын хүснэгтэд нэмнэ.
- Шахсан текст = 0122335

LZW шахалт

code	0	1	2	3	4	5	6	7	8	9
key	a	b	ab	ba	aba	abb	bab	baa	abba	abbaa

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 0122335
- p = abba
- pCode = 8
- c = a
- abba -г 8 -аар дүрслээд abbaa –г кодын хүснэгтэд нэмнэ.
- Шахсан текст = 01223358

LZW шахалт

code	0	1	2	3	4	5	6	7	8	9
key	a	b	ab	ba	aba	abb	bab	baa	abba	abbaa

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 01223358
- p = abba
- pCode = 8
- c = null
- abba -г 8 –аар дүрсэлнэ
- Шахсан текст = 012233588

Кодын хүснэгтийн дүрслэл

code	0	1	2	3	4	5	6	7	8	9
key	a	b	ab	ba	aba	abb	bab	baa	abba	abbaa

- Толь бичиг.
  - Хосууд (key, element) = (key,code).
  - Үйлдлүүд : get(key) , put(key, code)
- Кодын хязгаар 2^{12}.
- Хэш хүснэгт ашиглах.
  - Хувьсах урттай түлхүүрийг ижил ижил урттай болгох.
  - Түлхүүр бүр pc хэлбэртэй. Үүнд: p тэмдэгт мөр бол хүснэгтэд өмнө нь байгаа түлхүүр.
  - pc -г (pCode)c -оор сольно

Кодын хүснэгтийн дүрслэл

code	0	1	2	3	4	5	6	7	8	9
key	a	b	ab	ba	aba	abb	bab	baa	abba	abbaa

code	0	1	2	3	4	5	6	7	8	9
key	a	b	0b	1a	2a	2b	3b	3a	5a	8a

LZW задлалт

code	0	1
key	a	b

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- Кодыг зүүнээс баруун тийш текстэд хөрвүүлнэ
- 0 бол a.
- Задалсан текст = a
- pCode = 0 , p = a.
- p = a -ийн араас дараачийн тэмдэгтийг кодын хүснэгтэд оруулна.

LZW задлалт

code	0	1	2
key	a	b	ab

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- 1 бол b.
- Задалсан текст = ab
- pCode = 1 , p = b.
- lastP = a –ийн араас залгасан p –н эхний тэмдэгтийг кодын хүснэгтэд оруулна.

LZW задлалт

code	0	1	2	3
key	a	b	ab	ba

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- 2 бол ab.
- Задалман текст = abab
- pCode = 2 , p = ab.
- lastP = b –ийн араас залгасан p –н эхний тэмдэгтийг кодын хүснэгтэд оруулна.

LZW задлалт

code	0	1	2	3	4
key	a	b	ab	ba	aba

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- 2 бол ab
- Задалсан текст = ababab.
- pCode = 2 , p = ab.
- lastP = ab –ийн араас залгасан p –н эхний тэмдэгтийг кодын хүснэгтэд оруулна.

LZW задлалт

code	0	1	2	3	4	5
key	a	b	ab	ba	aba	abb

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- 3 бол ba
- Задалсан текст = abababba.
- pCode = 3 , p = ba.
- lastP = ab –ийн араас залгасан p –н эхний тэмдэгтийг кодын хүснэгтэд оруулна.

LZW задлалт

code	0	1	2	3	4	5	6
key	a	b	ab	ba	aba	abb	bab

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- 3 бол ba
- Задалсан текст = abababbaba.
- pCode = 3 , p = ba.
- lastP = ba –ийн араас залгасан p –н эхний тэмдэгтийг кодын хүснэгтэдоруулна.

LZW задлалт

code	0	1	2	3	4	5	6	7
key	a	b	ab	ba	aba	abb	bab	baa

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- 5 бол abb
- Задалсан текст = abababbabaabb.
- pCode = 5 , p = abb.
- lastP = ba –ийн араас залгасан p –н эхний тэмдэгтийг кодын хүснэгтэд оруулна.

LZW задлалт

code	0	1	2	3	4	5	6	7	8
key	a	b	ab	ba	aba	abb	bab	baa	abba

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- 8 бол ???
- Код хүснэгтэд байхгүй бол, түүний түлхүүр нь lastP –ий араас залгасан lastP-ий эхний тэмдэгт байх болно
- lastP = abb
- Тэгэхээр 8 бол abba.

LZW задлалт

code	0	1	2	3	4	5	6	7	8	9
key	a	b	ab	ba	aba	abb	bab	baa	abba	abbaa

- Эх текст = abababbabaabbabbaabba
- Шахсан текст = 012233588
- 8 бол abba
- Задалсан текст = abababbabaabbabbaabba.
- pCode = 8 , p = abba.
- lastP = abba –ийн араас орсон p –ийн эхний тэмдэгт кодын хүснэгтэд орно.

Кодын хүснэгтийн дүрслэл

code	0	1	2	3	4	5	6	7	8	9
key	a	b	ab	ba	aba	abb	bab	baa	abba	abbaa

- Толь бичиг.
  - Хосууд (key, element) = (code, what the code represents) = (code, codeKey).
  - Үйлдлүүд : get(key) , put(key, code)
- Түлхүүрүүд бүхэл тоо 0, 1, 2, …
- 1D codeTable массивыг ашиглая
  - codeTable[code] = codeKey.
  - Кодын түлхүүр бүр pc хэлбэртэй. Үүнд p тэмдэгт мөр бол хүснэгтэд өмнө нь орсон кодын түлхүүр.
  - pc -г (pCode)c -аар сольно

Хугацааны үзүүлэлт

- Шахалт.
  - O(n) дундаж хугацаа, үүнд n шахагдсан текстийн урт.
- Задлалт.
  - O(n) хугацаа, үүнд n задалсан текстийн урт.

- “Анхны тоонд хуваах үлдэгдэл” аргачлал нь бидний мэдэх хаашинг (hashing) алгоритм билээ. Энэ аргачлалд түлхүүр утга нь N тоонд хуваагддаг ба мөн түүний ижил хаашын утга хэмээн нэрлэгддэг үлдэгдэл нь шууд хаашын хүснэгтэнд индексээр хэрэглэгддэг. N бол хаяглаж болох зайны хэмжээтэй тэнцүү эсвэл хамгийн ойр дөхөх анхны тоо юм. Хэрэв 20 хаяглалт хийхүйц зайтай бол дараах сонголтуудын аль нь түлхүүр утга 136 байхад тооцоолж гарах хаашын утга вэ? Энд, анхны тоо гэдэг нь нэгээс бусад бүх тоонд тэгш хуваагддаггүй тоог хэлнэ. 2, 3, 5, 7, 11, ба 13 нь эхний 6 анхны тоо юм.
- A) 0 B) 1 C) 3 D) 16

- www.itpec.org