Activity:
Linear Hashing

File State

- The file state of an LH file is determined by the number n of buckets
  - level l
  - split pointer s
  - Formula: $n = 2^l + s$ with l as high as possible, i.e. with $s \in \{0,1,\ldots,2^l - 1\}$

File State

- Clarification regarding the literature
    - The original LH scheme can start with any number of buckets
    - In this class, we are using the most common case

Exercise

- What is the level and the state of an LH file with 13 buckets?

Solution

- We write $13 = 2^3 + 5$
  - Level is $l = \lfloor \log_2(13) \rfloor = 3$
  - Split pointer is $s = 13 - l$

Exercise

- Where would the records with the following (randomly picked) keys be inserted?
- 82
- 27
- 37

Solution

- Level is 3, so we use first remainder modulo $2^3 = 8$ and $2^4 = 16$ second

- $82 \pmod{8} = 2$. Since $2 < 5$, we rehash: $82 \pmod{16} = 2$ and we insert into bucket 2

- $27 \pmod{8} = 3$. Since 3 < 5, we rehash: $27 \pmod{16} = 11$. We insert into bucket 11

- $37 \pmod{8} = 5$. Since $5 \nless 5$, we do not rehash but insert into bucket 5.

Exercise

- Where would the records with the following (randomly picked) keys be inserted?
- 48
- 60
- 63
- 71

Solution

- $48 \pmod 8 = 0$. Rehash: $48 \pmod{16} = 0$ and insert into bucket 0.
- $60 \pmod 8 = 4$. Rehash: $60 \pmod{16} = 12$ and insert into bucket 12.
- $63 \pmod 8 = 7$. Rehash not necessary. Insert into bucket 7.
- $71 \pmod 8 = 7$. No rehash is necessary.

Exercise

- Where would the records with the following (randomly picked) keys be inserted?
- 98
- 75
- 25
- 30

Solution

- $98 \pmod 8 = 2$. Rehash: $98 \pmod{16} = 2$. Insert into bucket 2
- $75 \pmod 8 = 3$. Rehash: $75 \pmod{16} = 11$. Insert into bucket 11
- $25 \pmod 8 = 1$. Rehash: $25 \pmod{16} = 9$. Insert into bucket 9.
- $30 \pmod 8 = 6$. Insert into bucket 6.

Exercise

- Give the level and split pointer values as an LH file moves from 6 buckets to 20

Solution

Nr o Buckets	Level	Split Ptr
6	2	2
7	2	3
8	3	0
9	3	1
10	3	2
11	3	3
12	3	4
13	3	5
14	3	6
15	3	7
16	4	0
17	4	1
18	4	2
19	4	3
20	4	4

Interpretation

- We can encapsulate the behavior of the level and split pointer into the following algorithm

```
def split(level, split_pointer):
     split_pointer += 1
     if split_pointer == 2**level:
          split_pointer = 0
          level += 1
     return (level, split_pointer)
```

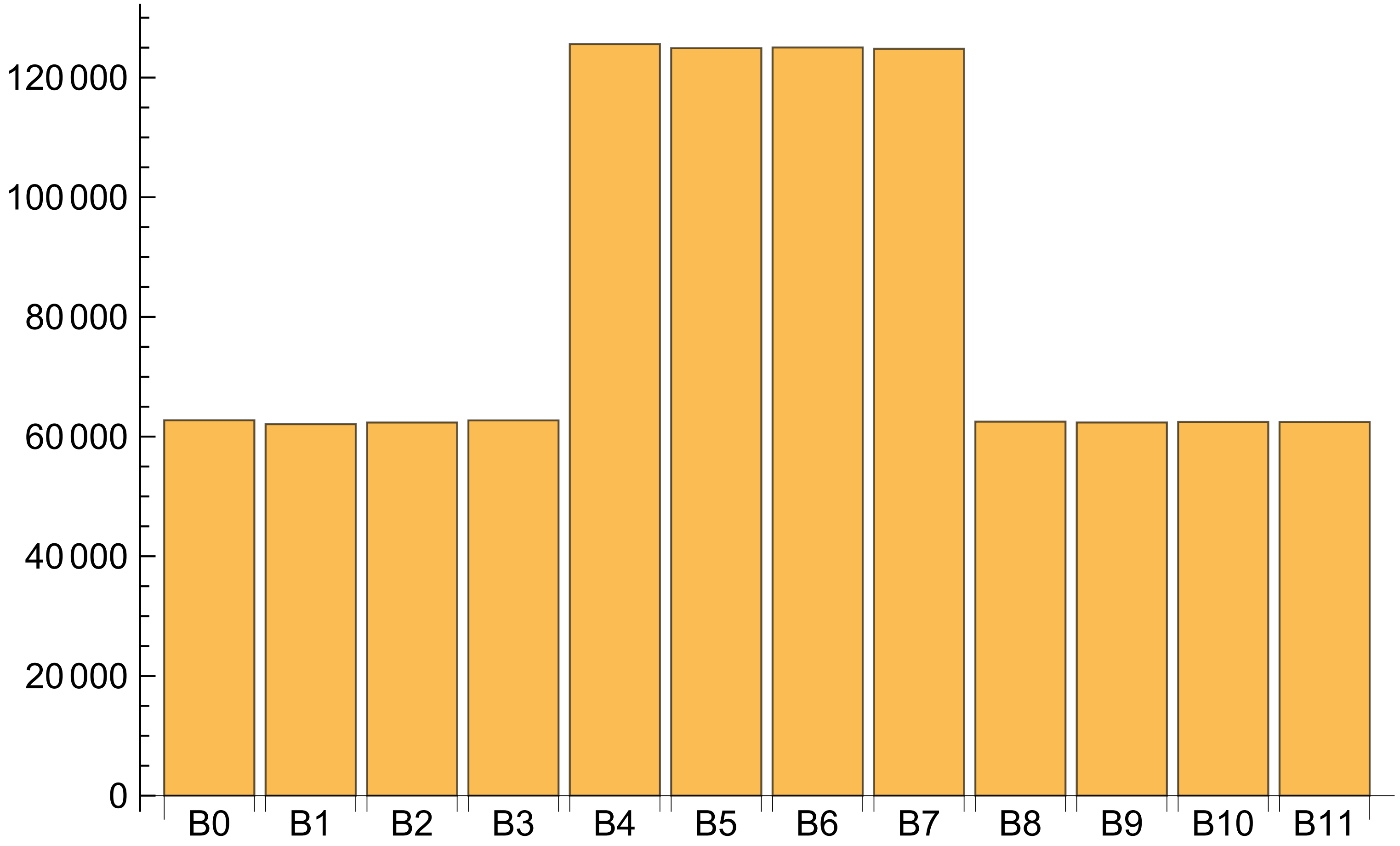
- We increment the split pointer
- If the split pointer equals 2^{level} then set the split pointer to zero and increment the level

Programming Exercise

- Using a programming platform of your choice, implement the LH addressing algorithm
- Insert 1000 records with key uniformly selected between 0 and $2^{32} - 1$ into an LH file with (a) 12 and (b) 25 buckets.
- Look at the size of the buckets.

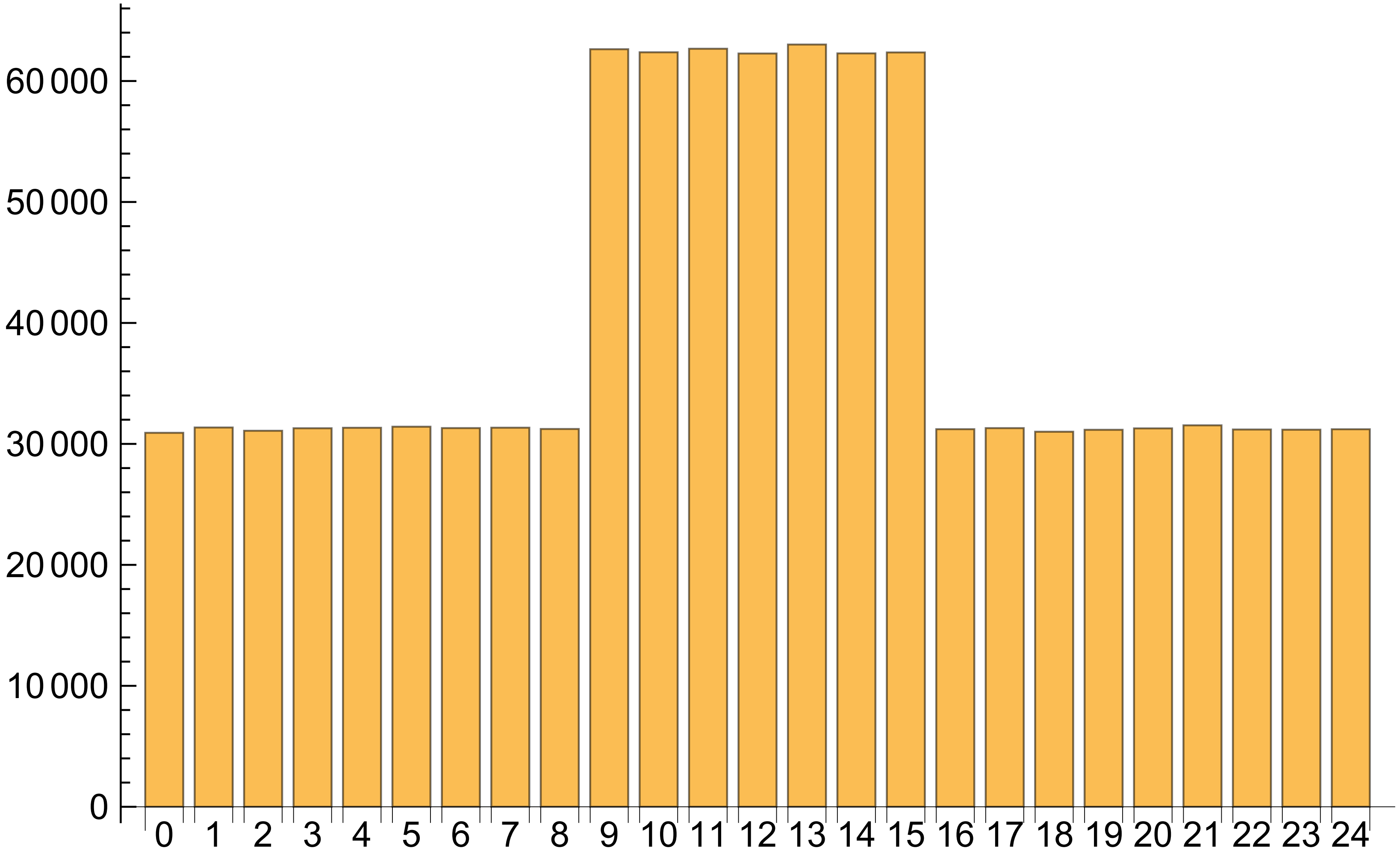
Solution

- I changed the number to 1,000,000
  - For 12 buckets:



Solution

- Here is the chart for 25 buckets



Interpretation

- Even with a perfect hash function, an LH file has buckets of equal size only if the number of buckets is a power of two.
- Otherwise, there are buckets already split in the current round and those not yet split.
    - The latter have about twice as many records