アルゴリズムとデータ構造

第8週目

担当　情報システム部門　徳光政弘
2025年6月18日

今日の内容

- ハッシュ法

ハッシュ法

- 効率的にデータを探索するためのデータ格納方法と検索方法
- 同じデータに対する衝突回避
  - チェイン法(次回)
  - オープンアドレス法(今回)

ハッシュ法の考え方

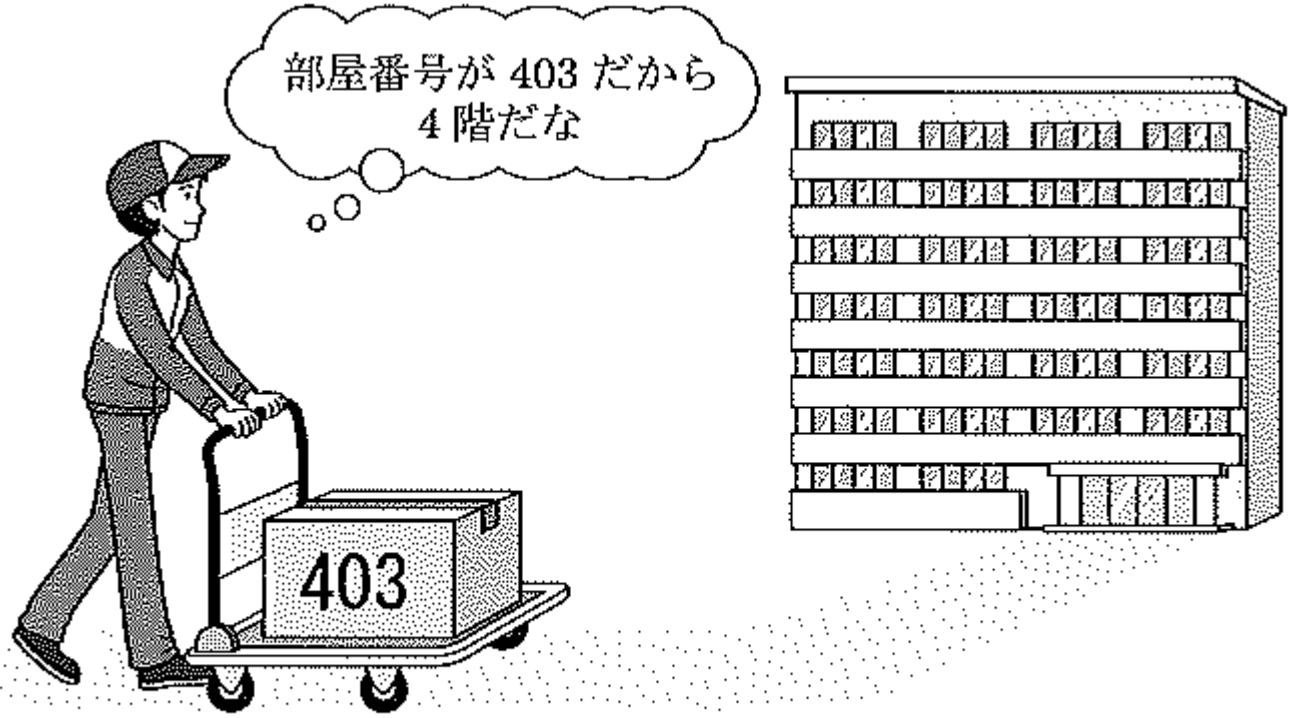


図 4.4　日常におけるハッシュ法のアイデア

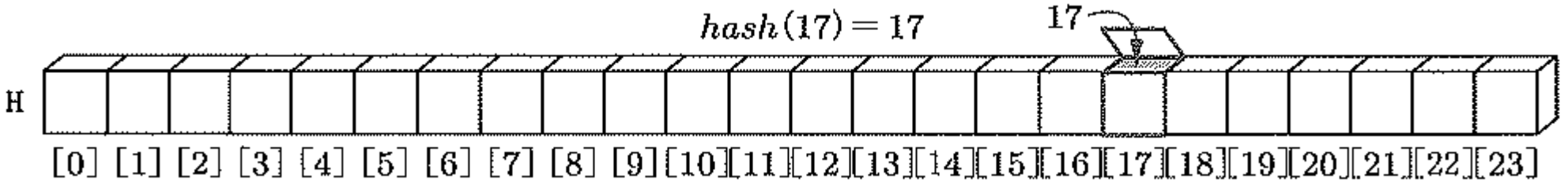
ハッシュ法の考え方

- 部屋番号がわかると、階数が推測できる
- 考え方　与えられたキーから場所を推測する
- 平均時間計算量

$$O(n)$$

ハッシュ法の考え方

- データxを格納する場所をハッシュ関数hash(x)で計算する
- 配列を使用する
- hash(x)はデータを格納する添字を返す



ハッシュ関数の定義

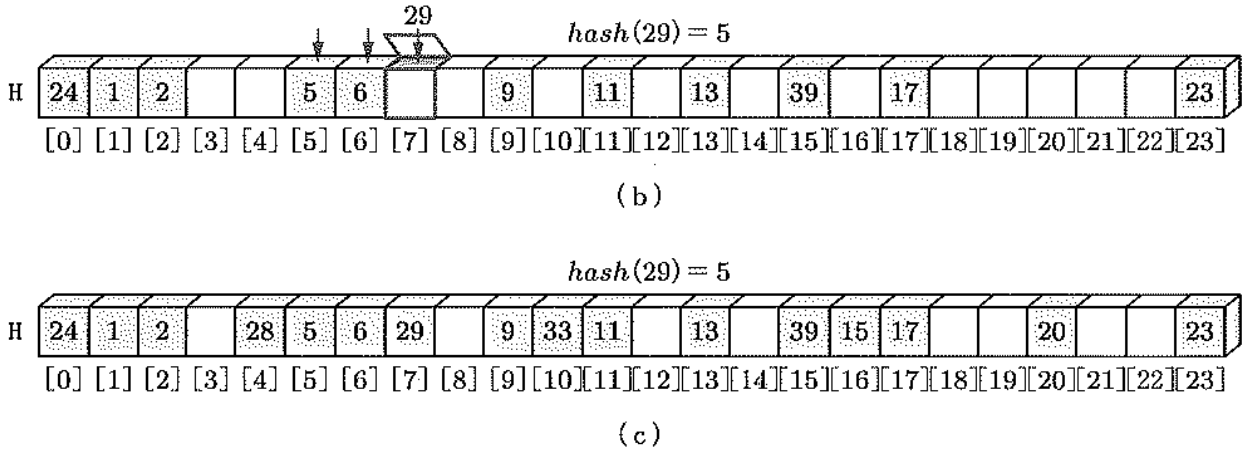
- ハッシュ関数は様々な定義(計算式)を考えることができる
- ハッシュ関数はSHAアルゴリズム、MD5でも使われる関数(定義は別)にもなっている、よく出てくる用語

$$hash(x) = (x \text{ を } 24 \text{ で割った余り})$$

教科書のハッシュ関数の定義

ハッシュのデータ挿入

- hash(29) = 5はデータがあるため、代入できない
- H[6]もデータがあるため、代入できない
- H[7]は空きがあるため、データを挿入できる



ハッシュのデータ挿入

アルゴリズム 4.3　ハッシュ法によるデータの格納

入力：サイズ m の配列 H, および n 個のデータを格納する配列 D

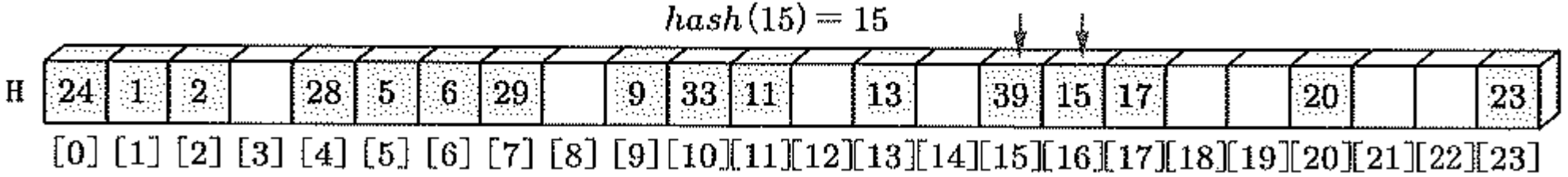
```
for (i=0; i<n; i=i+1) {
  k=hash(D[i]);
  while ( H[k]にデータが格納されている ) { k=((k+1)をm で割った余り); }
  H[k]=D[i];
}
```

k=((k+1) を m で割った余り);

余りは巡回するため、配列の末尾から先頭に添字を戻すことができる

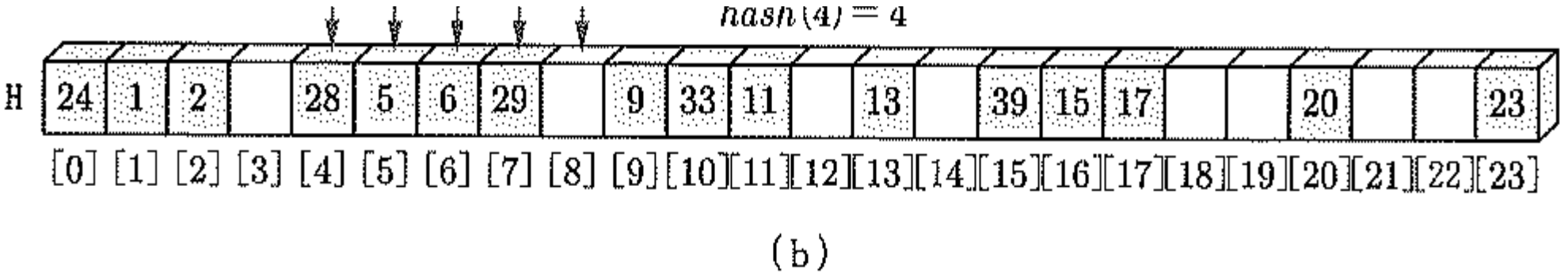
ハッシュのデータ探索

- hash(x)=15の値は39のため、H[16]の値が返される



ハッシュのデータ探索

- hash(4)=4を求めたが、H[4]、H[5]、H[6]と順番に調べて4が見つからない
- 探索に失敗した結果を返す



(b)

ハッシュのデータ探索

アルゴリズム 4.4　ハッシュ法による探索

入力：アルゴリズム 4.3 によりデータの格納されたサイズ m の配列 H と探索する値 x

```
k=hash(x);
while ( H[k]にデータが格納されている ) {
  if (H[k]==x) { H[k]を出力しアルゴリズムを終了; }
  k=((k+1)をmで割った余り);
}
"xは存在しない"と出力;
```

ハッシュ関数の時間計算量

- 平均時間計算量(証明は長く難しいため略)

●性質 4.1

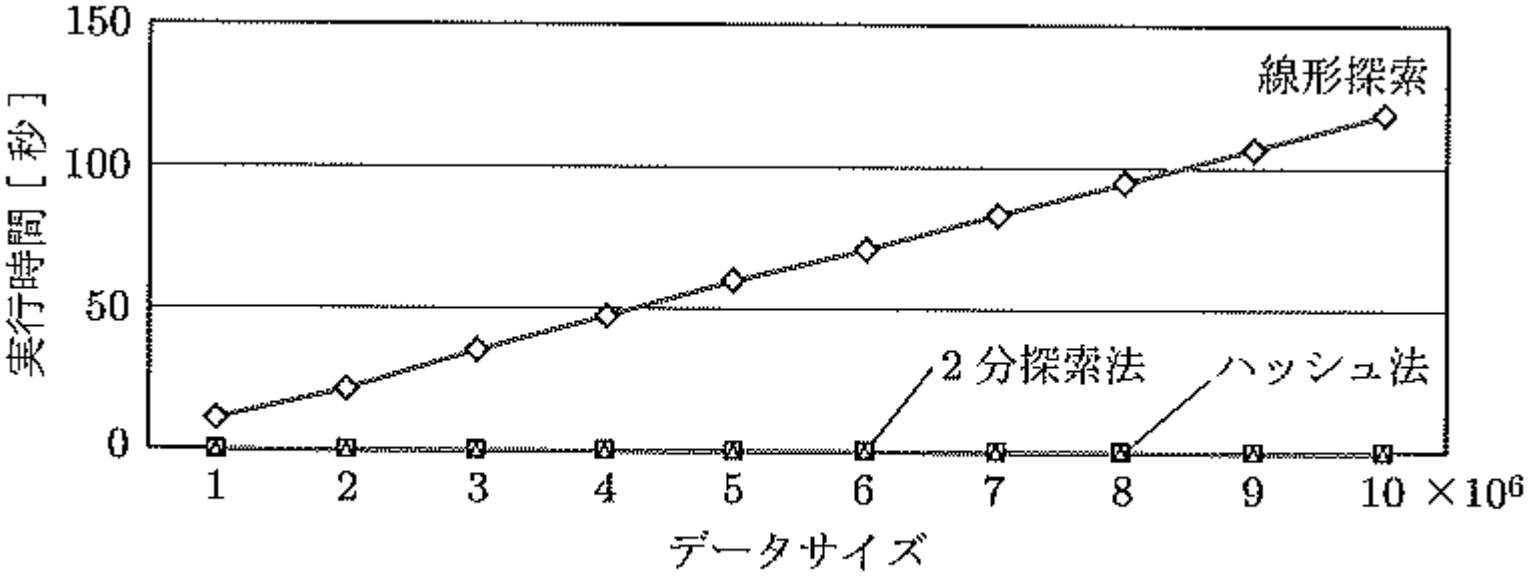
n 個のデータが，ハッシュ関数を用いてサイズが m の配列のランダムな場所に格納されているとする．このとき，ハッシュ法を用いた探索アルゴリズムの平均時間計算量は $O\left(\frac{m}{m-n}\right)$ である．

ハッシュ関数の最悪時間計算量

- 配列の大きさをデータ個数の1.5nとする
- 定数時間で探索できる

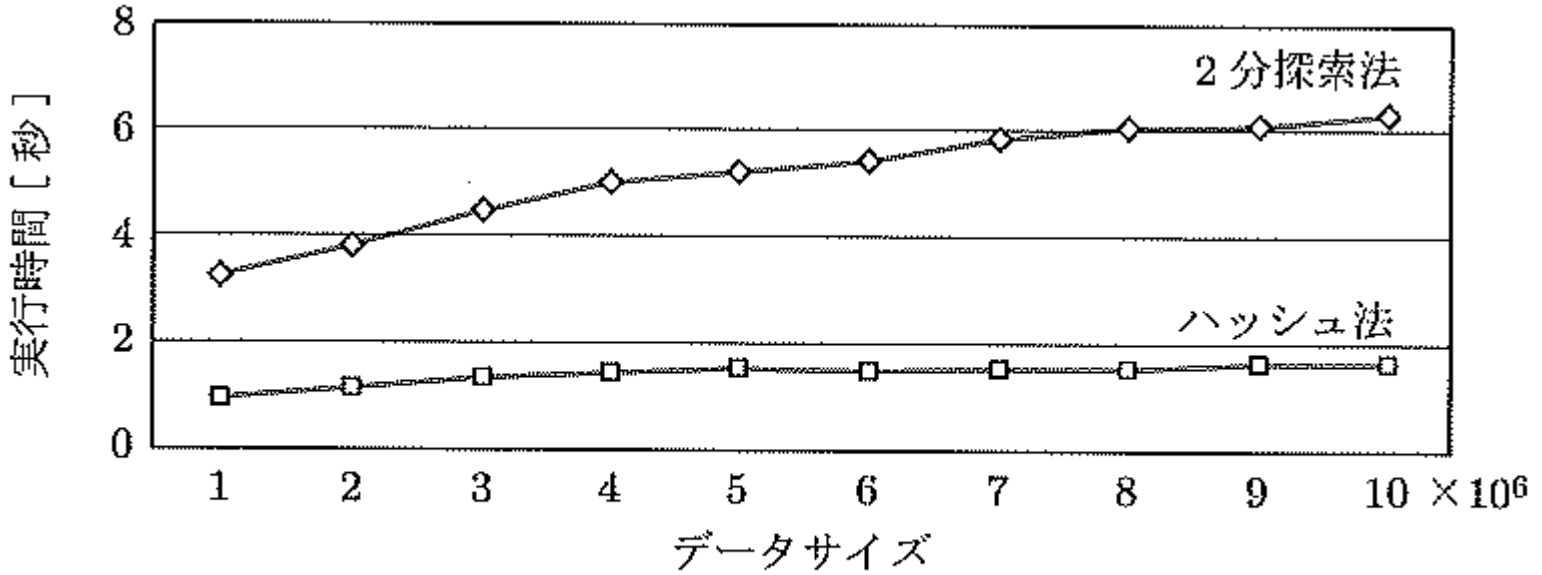
$$O\left(\frac{m}{m-n}\right) = O\left(\frac{1.5n}{1.5n-n}\right) = O(1)$$

探索アルゴリズムの実行速度比較



（a）線形探索，2 分探索法とハッシュ法の比較

探索アルゴリズムの実行速度比較



（b）2分探索法とハッシュ法の比較

理論上は2分探索法も高速な部類になるが、ハッシュ法はさらに高速な方法になっている