

# [B8] 二分探索とハッシュ

Copyright © 2020-2024, by Takeshi Kawabata

# 探索のアルゴリズム

- 大量のデータ中から必要な項目を探し出す
  - 線形探索      linear search
  - 二分探索      binary search
  - ハッシュ法    hash

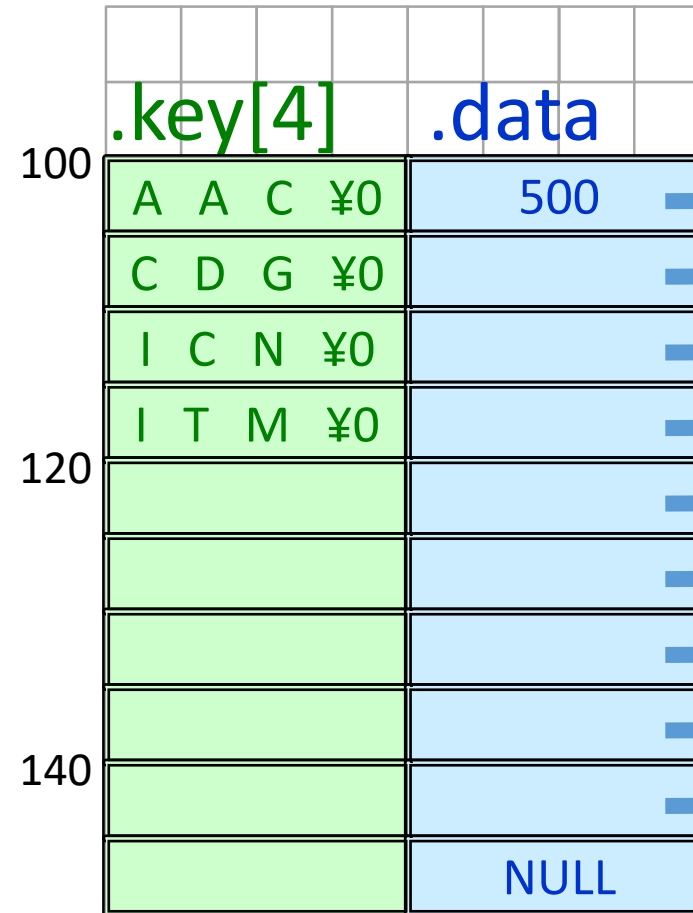
# データレコードのメモリ構造

```
#define KEY_LEN 3
#define MAX_RECORDS 10

typedef struct {
    char key[KEY_LEN+1];
    char *data; /* 後で malloc() */
} record_t;

record_t record[MAX_RECORDS];
```

プログラムのメモリ



オペレーティングシステムのヒープ領域



# 線形探索

探索キー

I T M ¥0

順に探す

プログラムのメモリ

|     | .key[4]  | .data |
|-----|----------|-------|
| 100 | A A C ¥0 | 500   |
|     | C D G ¥0 |       |
|     | I C N ¥0 |       |
| 120 | I T M ¥0 |       |
|     |          |       |
|     |          |       |
|     |          |       |
| 140 |          |       |
|     |          |       |
|     |          | NULL  |

オペレーティングシステムのヒープ領域

|  |                             |
|--|-----------------------------|
|  | Al Arish : Egypt – Al Arish |
|  | Paris : France Charles De   |
|  | Incheon [Seoul] : South K   |
|  | Osaka : Japan Itami Intern  |
|  |                             |
|  |                             |
|  |                             |
|  |                             |
|  |                             |

# 二分探索

探索キー

I T M ¥0

半分・  
半分と  
近づく

プログラムのメモリ

|     | .key[4]  | .data |
|-----|----------|-------|
| 100 | A A C ¥0 | 500   |
|     | C D G ¥0 |       |
|     | I C N ¥0 |       |
| 120 | I T M ¥0 |       |
|     |          |       |
|     |          |       |
|     |          |       |
| 140 |          |       |
|     |          |       |
|     |          | NULL  |

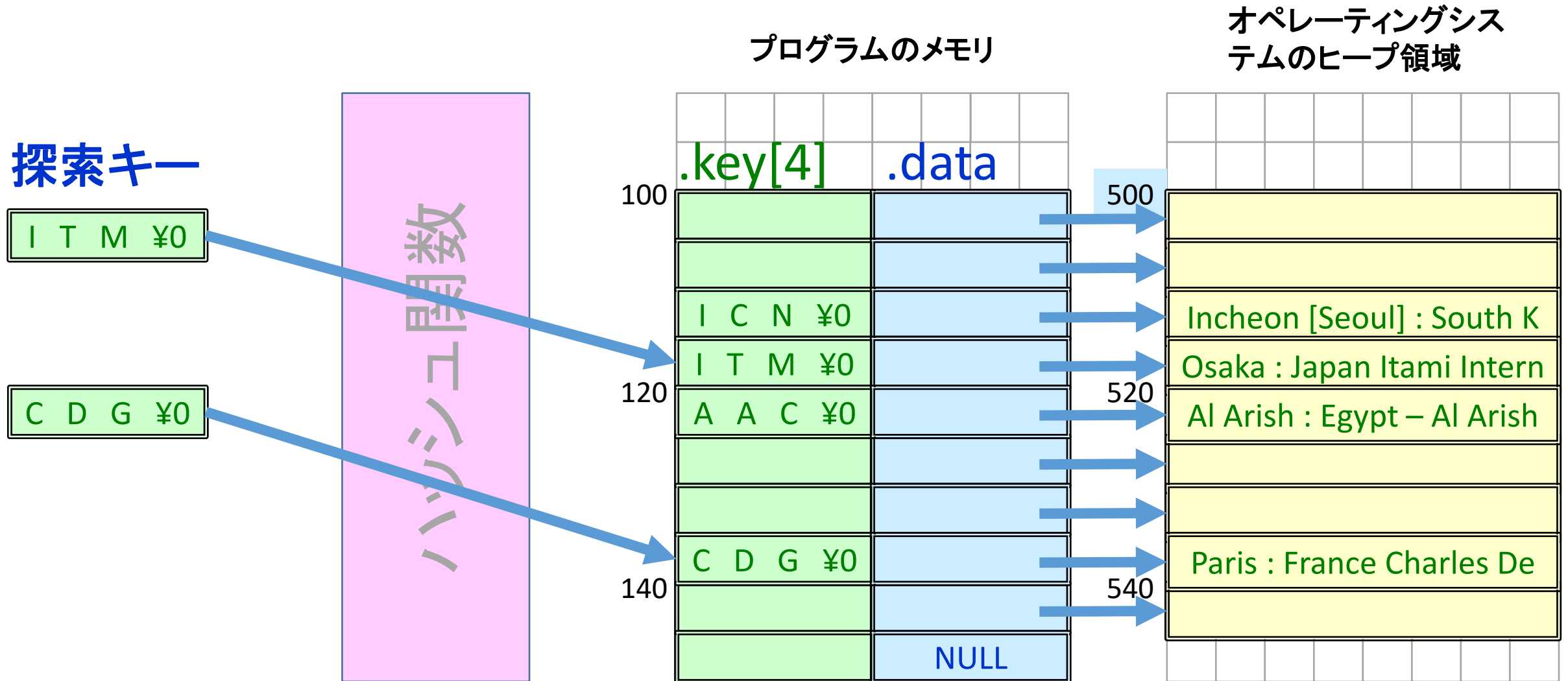
オペレーティングシステムのヒープ領域

|  |                             |
|--|-----------------------------|
|  | Al Arish : Egypt – Al Arish |
|  | Paris : France Charles De   |
|  | Incheon [Seoul] : South K   |
|  | Osaka : Japan Itami Intern  |
|  |                             |
|  |                             |
|  |                             |
|  |                             |
|  |                             |

# ハッシュ法

- ハッシュ関数
  - 検索キーを、より少ないビット数の「ハッシュ値」に写像
  - 異なるキーが同じハッシュ値に ⇒ 「衝突 (collision)」
- 衝突が発生した場合の対処
  - クローズドハッシュ法 (オープンアドレス法)  
(例えば、)後続するレコードを利用
  - オープンハッシュ法  
ハッシュ値ごとに線形リストを用意 (オプション課題 8)

# ハッシュ法



## ハッシュ値の衝突 (クローズドハッシュ法)

# プログラムのメモリ

## オペレーティングシステムのヒープ領域

