

4. 21, 2, 35, 14, 17, 5, 6, 20 というデータを順に与えたときに出来上がる二分探索木と、平衡が崩れた際に再並行化を行って作成した AVL 木を図示し、成功探索時の平均探索回数をそれぞれ求めなさい。(10 点)

5. 二分探索木を用いてデータの整列を行う方法について説明しなさい。(5 点)

50

6. 以下のハッシュ表の探索アルゴリズムが発生する番地(h の値)の系列 h_i ($i=0,1,\dots$) を, a, b および, i を用いて表しなさい。(5 点)

```

#include <stdio.h>
#include <string.h>
#define M 300

char *tab[M];

int hash0(char *K) { 未定義 }

int quadratic_search(char *K, int a, int b)
{
    int h,u,v;
    h = hash0(K);
    u = a+b; v = 2*b;
    while (tab[h] != NULL)
        if (strcmp(K, tab[h]) == 0) return h;
        else {h = (h+u)%M; u = u+v;}
    return -1;
}

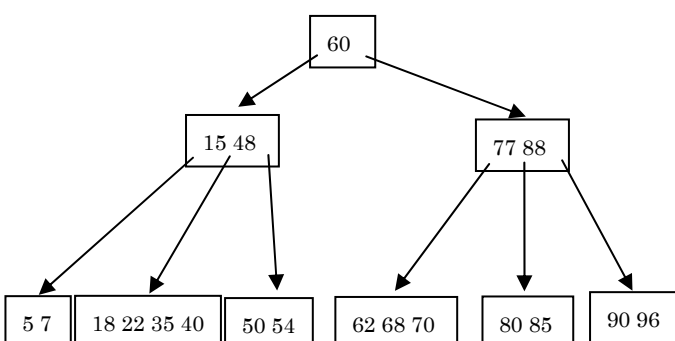
```

7. テキスト ABCABABBBBABABBA から文字列 ABABBA を見つけるための文字比較回数は, 単純照合法, KMP 法を用いた場合でそれぞれどのようになるか答えなさい。(10 点)

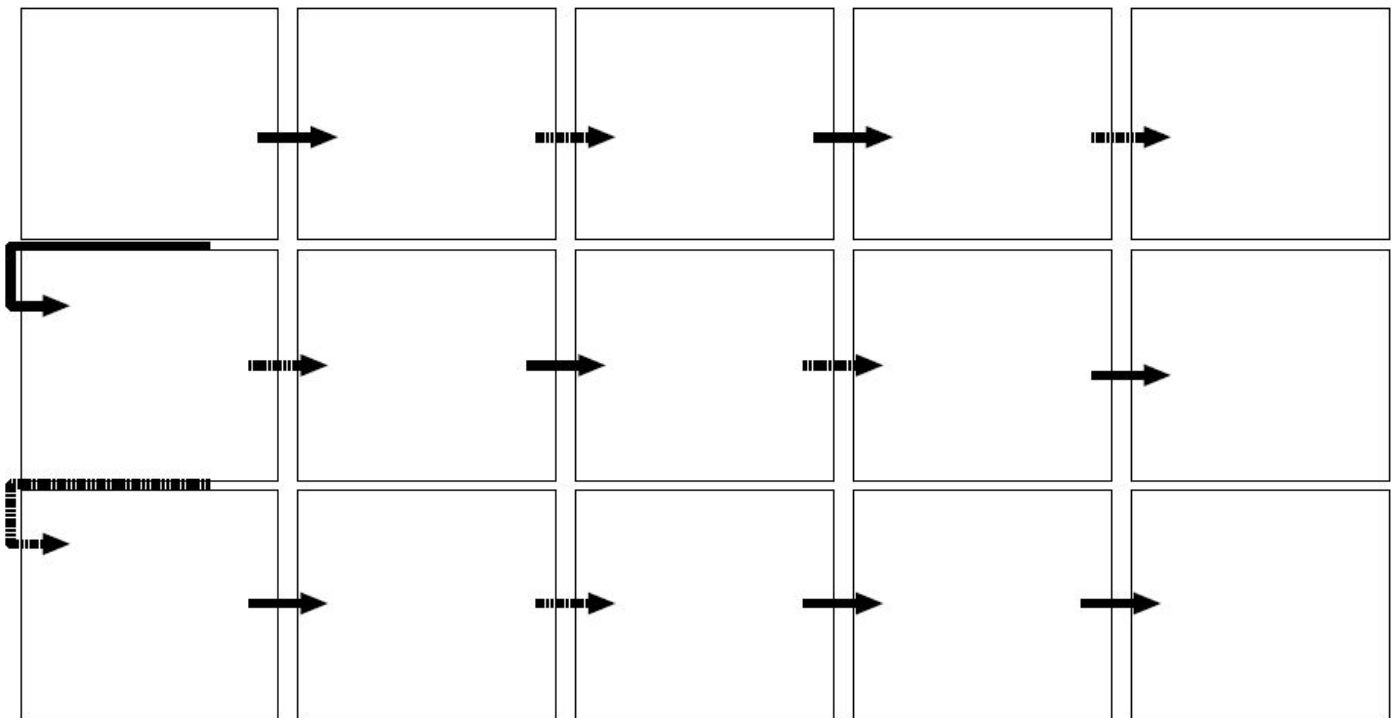
単純照合法: 回

KMP 法 回

8. 下に示す 2 次の B 木において, データ 88 を削除すると, 構造はどのように変化するか図示しなさい。(10 点)



9. 符号付き 8bit 整数のキー列, $-1\ 2\ 7, -5\ 0, 5\ 0, 6, 9$ をトライに格納した場合と, Patricia 木に格納した場合の木構造を示しなさい. 但し, 負の数の表現には 2 の補数を用いるものとする. (10 点)
10. キーが $5, 15, 19, 1, 6, 11, 10, 9, 14$ の順に配列 $A[1] \sim A[9]$ に格納されているものとする. この配列が順配置された完全 2 分木である場合, その木を図示しなさい. また, これをヒープに変更した場合の木構造を示しなさい. (5 点)
11. 上の問題 14 で得られたヒープを用いてヒープソートを行う場合の木構造の変化を, 最初のヒープの段階から図示しなさい. (10 点)



KMP 法の next[i]関数に関する説明の誤りについて.

先週の講義で, KMP 法の next[i]関数の説明が間違っていました. 下記のとおり修正致します.

next[i]は次のように定義される (図 3・4 参照).

$$\text{next}[i] = \max |s| \{ s \geq 1 \text{ かつ } \text{pat}[1 \dots (s-1)] = \text{pat}[(i-s+1) \dots (i-1)] \text{ かつ } \text{pat}[s] \neq \text{pat}[i] \} \text{ または } \{s=0\}$$

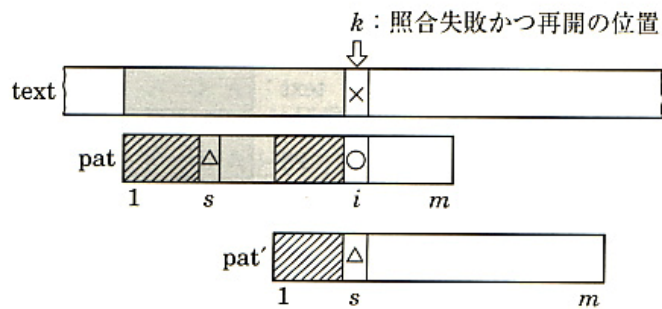


図 3・4 pat の移動位置

- 左図のように共通領域が存在し, $\text{pat}[s] \neq \text{pat}[i]$ となるケース: $\text{next}[i]=s$
- 共通領域の長さが 0 で, $\text{pat}[s] \neq \text{pat}[i]$ となるケース: $\text{next}[i]=1$
- 共通領域の長さが 0 で, $\text{pat}[s]=\text{pat}[i]$ となるケース: $\text{next}[i]=0$

実例

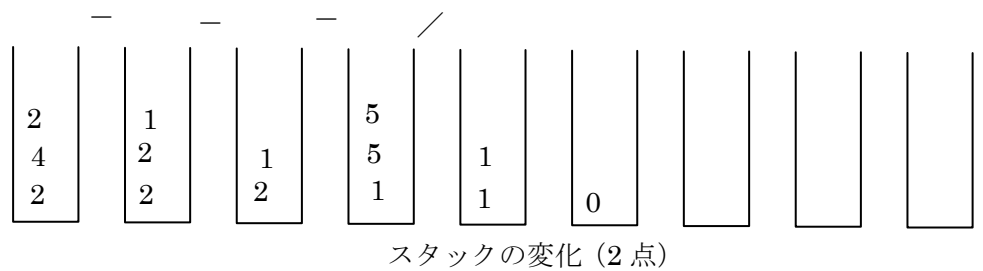
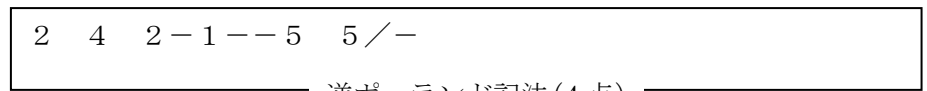
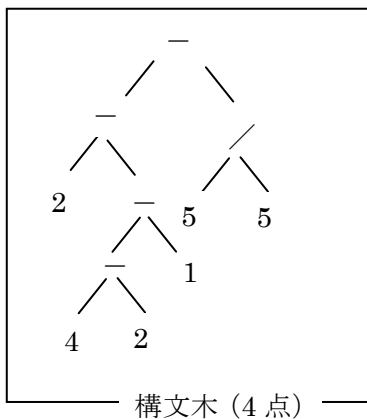
ABABBA
0 1 0 1 3 0

ABCDABD
0 1 1 1 0 1 3

先週の説明では, $\text{next}[i]=s$ となるケースについてのみの説明でした.

1. 以下の文章の空欄を埋めなさい。(各1点：計15点)

- 表探索において、最もよく用いられるのが **ハッシュ法** である。この方法は、探索を行うキーの値からデータの格納されるべき場所 (**ハッシュ 番地**) を直接計算する方法で、クラスタが生じてない状態では高速に探索が行える。しかし、表サイズに対する格納キーの個数の割合 (**表占有 率**) が増加すると、衝突処理が行われる確率が増え、探索効率が低下する。
 - 衝突処理は開番地法と **連鎖法** に分類されるが、開番地法の一つである **線形** 走査法では、表サイズとハッシュ増分が互いに素である場合、探索周期は **全表的** になる。
 - キーを2進数で表記したときの各桁の値を辺に付けた二分木構造は **トライ** と呼ばれる。この木構造の中には、一般に分岐の無い節が含まれるので、次にチェックすべき桁の位置を節に格納することによって、非分岐節を無くしたものが **Patricia 木** である。
 - k 次の B 木は、根には **1** 個以上 **2k** 個以下のキーが格納され、それ以外の節には **k** 個以上 **2k** 個以下のキーが格納され、全ての葉は同一レベルにあるというデータ構造である。
 - KMP** 法は、テキストから文字列パターンを高速に見つけ出す方法であり、文字列パターンの構造を予め調べておき、照合失敗時にできるだけパターンを大きくずらす方法である。**BM** 法は、このアイデアに加えて照合に失敗したテキスト側の文字が文字列パターンの照合失敗位置の左に現れる位置を調べておくことで、ずらしの量を決定する方法である。但し、日本語のように文字種の多い文字列に対しては **BM** 法は適用することが困難である。
2. 数式 $2 - (4 - 2 - 1) - 5 / 5$ の演算子木 (構文木) を示しなさい。また、この木を後順走査して得られる逆ポーランド記法を示しなさい。さらにこの記法の表す内容を計算する際のスタックの変化を図示しなさい。



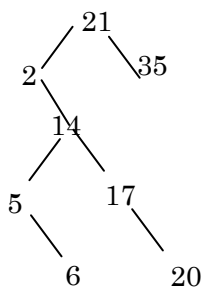
3. 2, 5, 10, 3, 15, 19, 14, 11, 8, 1 を順に与えたときのハッシュ表と成功探索時の平均探索回数を求めなさい。但し、表サイズは 11, $h_0(x) = x \bmod 11, h_i(x) = (h_0(x) + 2 \times i) \bmod 11$ とする。(10点)

表	11	8	2	3	15	5		14	19	1	10
探索回数	1	3	1	1	1	1		3	1	5	1

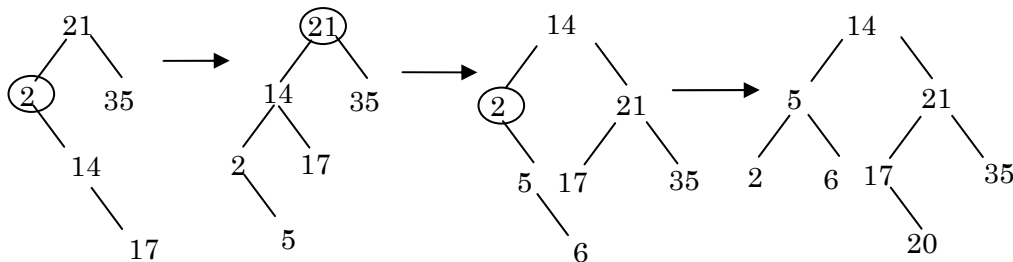
平均探索回数 = $18/10 = 1.8$ 回

合計 18 回

4. 21, 2, 35, 14, 17, 5, 6, 20 というデータを順に与えたときに出来上がる二分探索木と, 平衡が崩れた際に再並行化を行って作成した AVL 木を図示し, 成功探索時の平均探索回数をそれぞれ求めなさい. (10 点)



平均探索回数 $26/8=3.25$ 回 (2 点)



平均探索回数 $21/8=2.625$ 回 (8 点)

5. 二分探索木を用いてデータの整列を行う方法について説明しなさい. (5 点)

二分探索木を中順走査すればよい.

6. 以下のハッシュ表の探索アルゴリズムが発生する番地(h の値)の系列 h_i ($i=0,1,\dots$) を, a, b および, i を用いて表しなさい. (5 点)

```
#include <stdio.h>
#include <string.h>
#define M 300

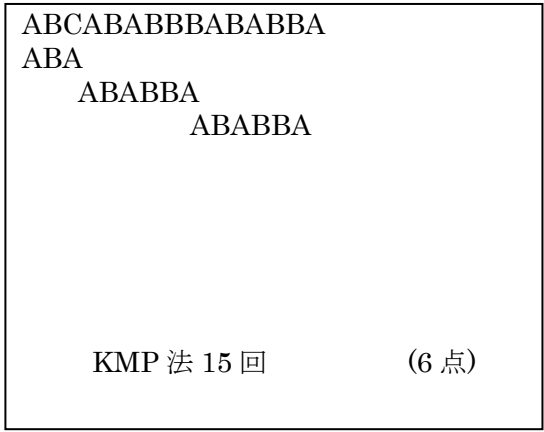
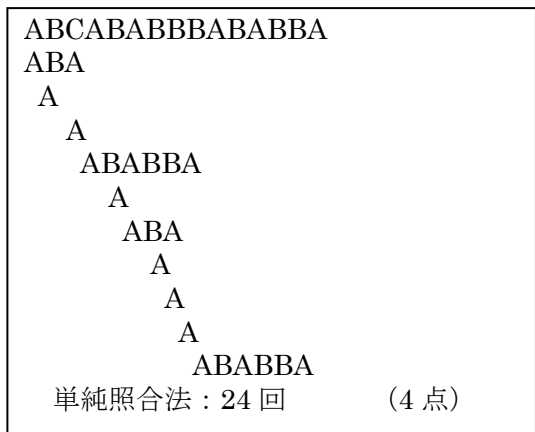
char *tab[M];

int hash0(char *K) { 未定義 }

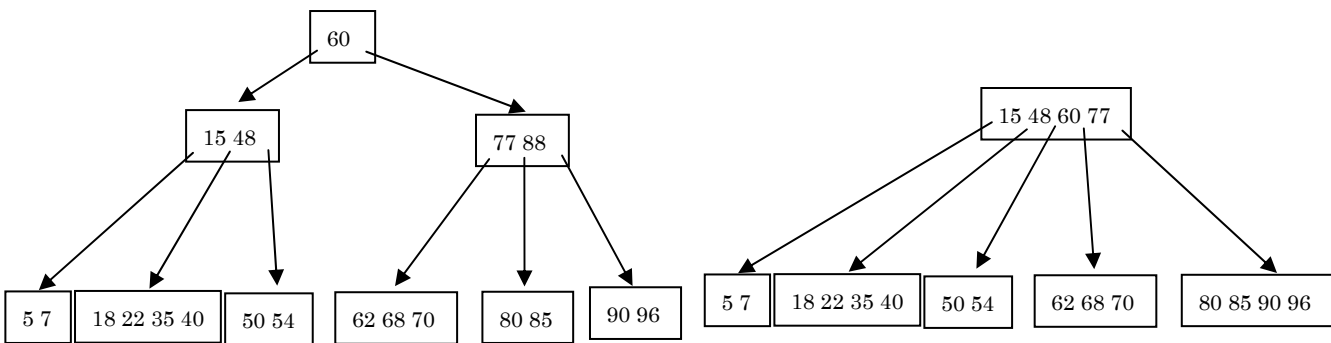
int quadratic_search(char *K, int a, int b)
{
    int h,u,v;
    h = hash0(K);
    u = a+b; v = 2*b;
    while (tab[h] != NULL)
        if (strcmp(K, tab[h]) == 0) return h;
        else {h = (h+u)%M; u = u+v;}
    return -1;
}
```

これは 2 次走査法である。
 $h_i(K) = (\text{hash0}(K) + a \cdot i + b \cdot i^2) \text{MOD } M$

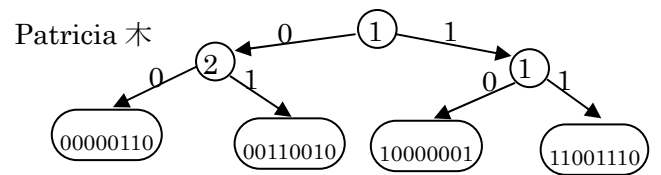
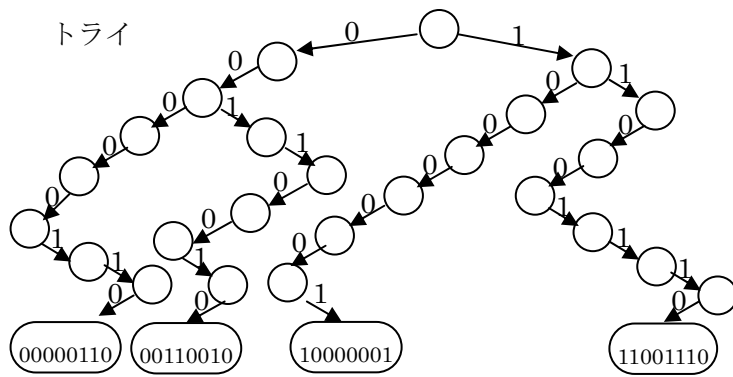
7. テキスト ABCABABBBBABABBA から文字列 ABABBA を見つけるための文字比較回数は, 単純照合法, KMP 法を用いた場合でそれぞれどのようになるか答えなさい. (10 点)



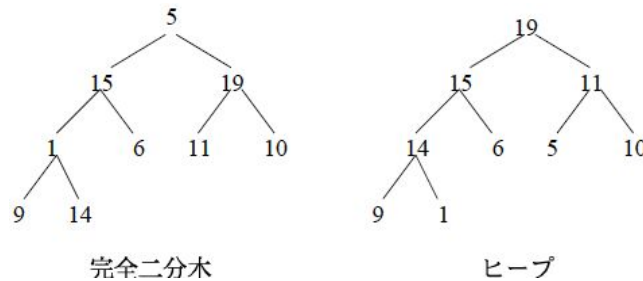
8. 下に示す 2 次の B 木において, データ 88 を削除すると, 構造はどのように変化するか図示しなさい. (10 点)



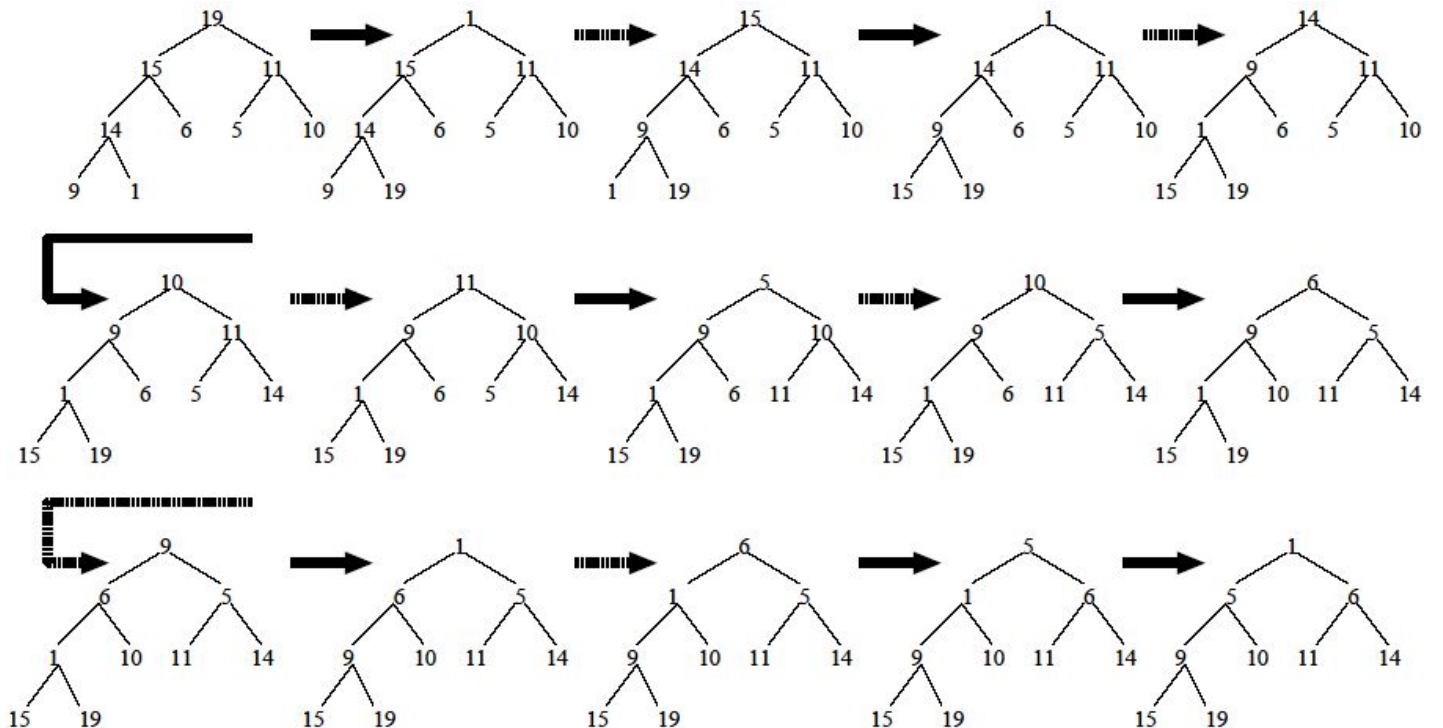
9. 符号付き 8bit 整数のキー列, $-127, -50, 50, 6, 9$ をトライに格納した場合と, Patricia 木に格納した場合の木構造を示しなさい. 但し, 負の数の表現には 2 の補数を用いるものとする. (10 点)



10. キーが $5, 15, 19, 1, 6, 11, 10, 9, 14$ の順に配列 $A[1] \sim A[9]$ に格納されているものとする. この配列が順配置された完全二分木である場合, その木を図示しなさい. また, これをヒープに変更した場合の木構造を示しなさい. (2 点+3 点: 5 点)



11. 上の問題 14 で得られたヒープを用いてヒープソートを行う場合の木構造の変化を, 最初のヒープの段階から図示しなさい. (10 点)



KMP 法の next[i]関数に関する説明の誤りについて.

先週の講義で, KMP 法の next[i]関数の説明が間違っていました. 下記のとおり修正致します.

next[i]は次のように定義される (図 3・4 参照).

$$\text{next}[i] = \max |s| \{ s \geq 1 \text{ かつ } \text{pat}[1 \dots (s-1)] = \text{pat}[(i-s+1) \dots (i-1)] \text{ かつ } \text{pat}[s] \neq \text{pat}[i] \} \text{ または } \{s=0\}$$

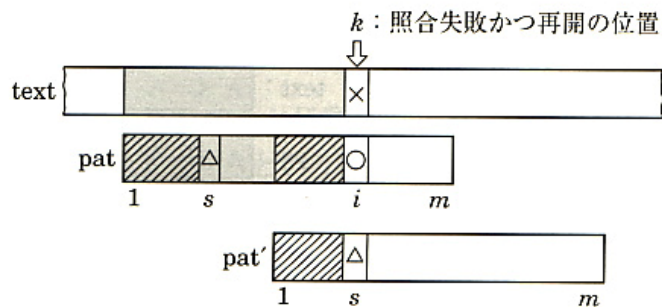


図 3・4 pat の移動位置

- 左図のように共通領域が存在し, $\text{pat}[s] \neq \text{pat}[i]$ となるケース: $\text{next}[i]=s$
- 共通領域の長さが0で, $\text{pat}[s] \neq \text{pat}[i]$ となるケース: $\text{next}[i]=1$
- 共通領域の長さが0で, $\text{pat}[s]=\text{pat}[i]$ となるケース: $\text{next}[i]=0$

実例

A	B	A	B	B	A
0	1	0	1	3	0

A	B	C	D	A	B	D
0	1	1	1	0	1	3

先週の説明では, next[i]=s となるケースについてのみの説明でした.