

Homework 6

- 给定字符串 $T = \text{BABCDABCDABEABCDABD}$; $P = \text{ABCDABD}$, 请给出 P 的 next 表, 并简要给出匹配过程 (为了简单, 可以给出匹配过程中 字符匹配失败且 $\text{next}[j]$ 不为 0 时 T, P, i, j 的状态)。

- **Solution**

next 表

A	B	C	D	A	B	D
-1	0	0	0	0	1	2

1.

B	A	B	C	A	B	C	D	A	B	C	D	A	B	E	A	B	C	D	A	B	D
				A	B	C	D	A	B	D											

$j = 6, i = 10$

$\text{next}[j] = \text{next}[6] = 2$

移动位数: $6 - 2 = 4$

2.

B	A	B	C	A	B	C	D	A	B	C	D	A	B	E	A	B	C	D	A	B	D
								A	B	C	D	A	B	D							

$j = 6, i = 14$

$\text{next}[j] = \text{next}[6] = 2$

移动位数: $6 - 2 = 4$

3.

B	A	B	C	A	B	C	D	A	B	C	D	A	B	E	A	B	C	D	A	B	D
															A	B	C	D	A	B	D

$j = 0, i = 14$

$\text{next}[j] = \text{next}[0] = -1$

匹配成功

2. 编写一段程序, 随机生成 10^4 , 10^5 , 10^6 个随机数, 并分别在这三者中, 分别使用快排和 `linearSelect()` 方法, 选择出第 100 大的数字 (需给出), 并给出三者的运行时间 (可以表格形式给出), 给出 `linearSelect()` 关键代码并解释。

	10^4	10^5	10^6
quickSort	0.001 (s)	0.011 (s)	0.1555
linearSelect	0 (s)	0.004 (s)	0.0481
the 100th largest num	9907	99899	999898

```

int linearSelect(int data[], int low, int high, int k) {
    //init
    int index, median, num_of_less_Median, size;
    size = high - low;

    // 第0步
    if (size < Q) {
        trivialSelect(data, low, high);
        return data[low + k - 1];
    }

    //第一步 划分
    //第二步 排序, 计算中位数, 调整中位数位置
    for (int i = 0; i < size / Q; i++) {
        trivialSelect(data, low + i * Q, low + i * Q + Q);
        swap(&data[low + i], &data[low + i * Q + 2]);
    }

    //第三步 递归找全局中位数
    median = linearSelect(data, low, low + size / Q, size / 10 + 1);

    //第四步 划分子集
    index = Partition(data, low, high, median);
    num_of_less_Median = index - low + 1;
    if (num_of_less_Median == k)
        return median;
    else if (num_of_less_Median > k)
        return linearSelect(data, low, index + 1, k);
    else
        return linearSelect(data, index + 1, high, k - num_of_less_Median);
}

```