

第9章 指针和数组

——指针数组及其在字符串处理中的应用

问题的提出

- 问题：存储一个字符串集合的最佳方式是什么？
 - 二维的字符数组

```
char country[][10] = {"America", "England", "Australia",  
                      "China", "Finland"};
```

<div>5</div>	A	m	e	r	i	c	a	\0	\0	\0
	E	n	g	l	a	n	d	\0	\0	\0
	A	u	s	t	r	a	l	i	a	\0
	C	h	i	n	a	\0	\0	\0	\0	\0
	F	i	n	l	a	n	d	\0	\0	\0
<div>10</div>										

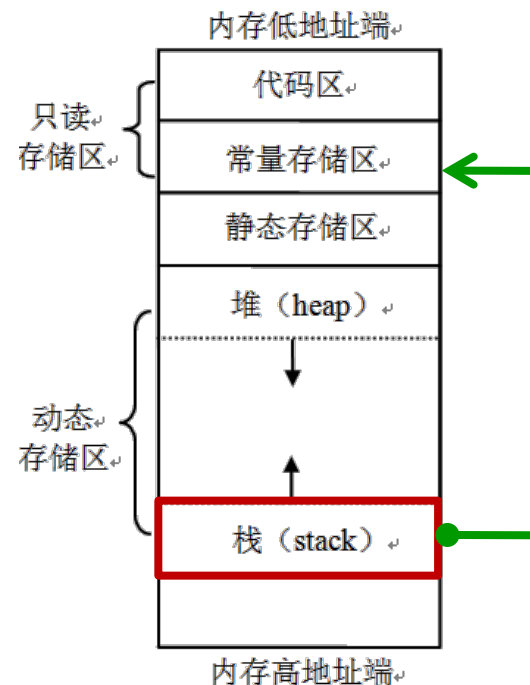
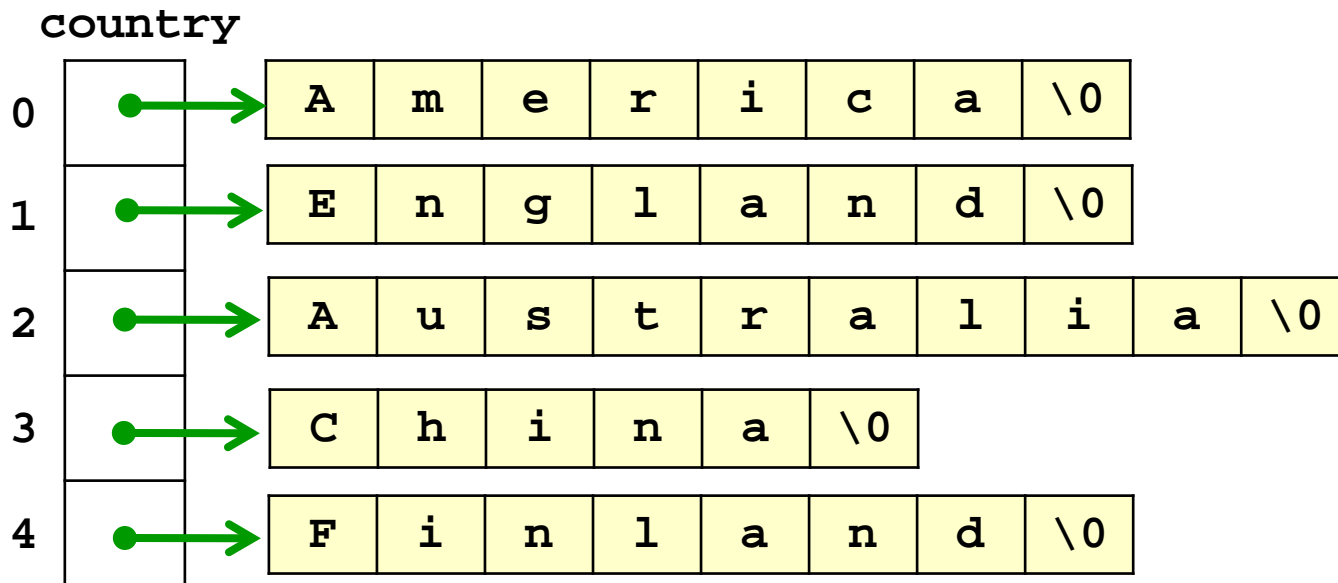
参差不齐的数组类型？



问题的解决

- 问题：存储一个字符串集合的最佳方式是什么？
 - 字符指针数组——每个数组元素都是一个指向字符串的指针

```
char *country[] = {"America", "England", "Australia",  
                  "China", "Finland"};
```



问题的解决

- 问题：如何访问指针数组指向的字符串？
 - 如何访问二维字符数组中的字符串？

```
char country[][10] = {"America", "England", "Australia",  
                      "China", "Finland"};
```

*(country+0)即country[0]

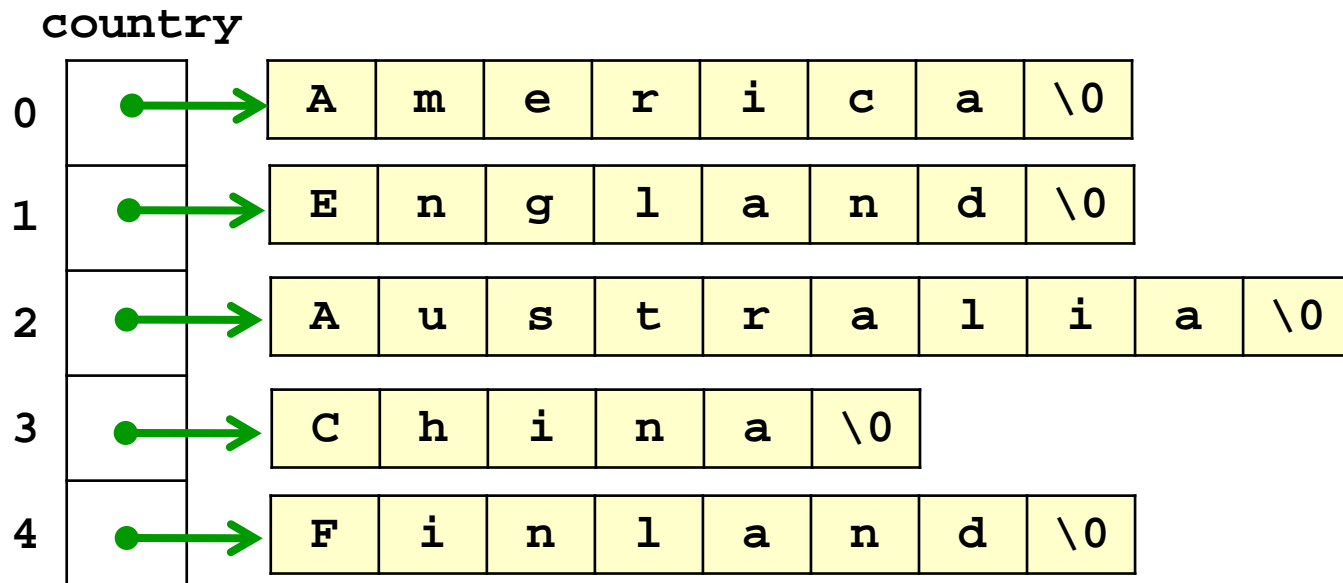
country	A	m	e	r	i	c	a	\0	\0	\0
country+1	E	n	g	l	a	n	d	\0	\0	\0
	A	u	s	t	r	a	l	i	a	\0
	C	h	i	n	a	\0	\0	\0	\0	\0
	F	i	n	l	a	n	d	\0	\0	\0

```
for (i=0; i<5; i++)  
{  
    printf("%s\n", country[i]);  
}
```

问题的解决

- 问题：如何访问指针数组指向的字符串？
 - 和访问二维字符数组中的字符串的方式相同

```
char *country[] = {"America", "England", "Australia",  
                  "China", "Finland"};
```



```
for (i=0; i<5; i++)  
{  
    printf("%s\n", country[i]);  
}
```

回头看——指针、数组及其他类型的混合

- 基本数据类型

- * `int`、`long`、`char`、`short`、`float`、`double`.....

- 数组是一种从其他类型派生的数据类型

- * 每个元素都有一个类型，称为数组的基类型

- 指针是一种从其他类型派生的数据类型

- * 指向X型变量的指针，X称为指针的基类型

- 任何类型都可作为指针或数组的基类型

- * 一个类型派生出新的类型，新的类型又派生出新的类型，

指针数组与指向数组的指针

- 用数组作为指针的基类型——指向数组的**指针**

- * 例如: `int (*p)[5];`

`p` \longrightarrow `*` \longrightarrow `int [5]`

- 用指针作数组的基类型——**指针数组** (Pointer Array)

- * 元素为指针类型的数组

- 定义形式为: **数据类型** `*数组名[数组长度];`

- * 例如: `char *p[5];`

`p` \longrightarrow `[5]` \longrightarrow `char *`

字符串的排序

按字典顺序对国名排序

```
void SortString(char str[][N], int n)
{
    int i, j;
    char temp[N];
    for (i=0; i<n-1; i++)
    {
        for (j=i+1; j<n; j++)
        {
            if (strcmp(str[j], str[i]) < 0)
            {
                strcpy(temp, str[i]);
                strcpy(str[i], str[j]);
                strcpy(str[j], temp);
            }
        }
    }
}
```

交换字符数组中的字符串

A	m	e	r	i	c	a	\0	\0	\0
E	n	g	l	a	n	d	\0	\0	\0
A	u	s	t	r	a	l	i	a	\0
C	h	i	n	a	\0	\0	\0	\0	\0
F	i	n	l	a	n	d	\0	\0	\0



物理排序

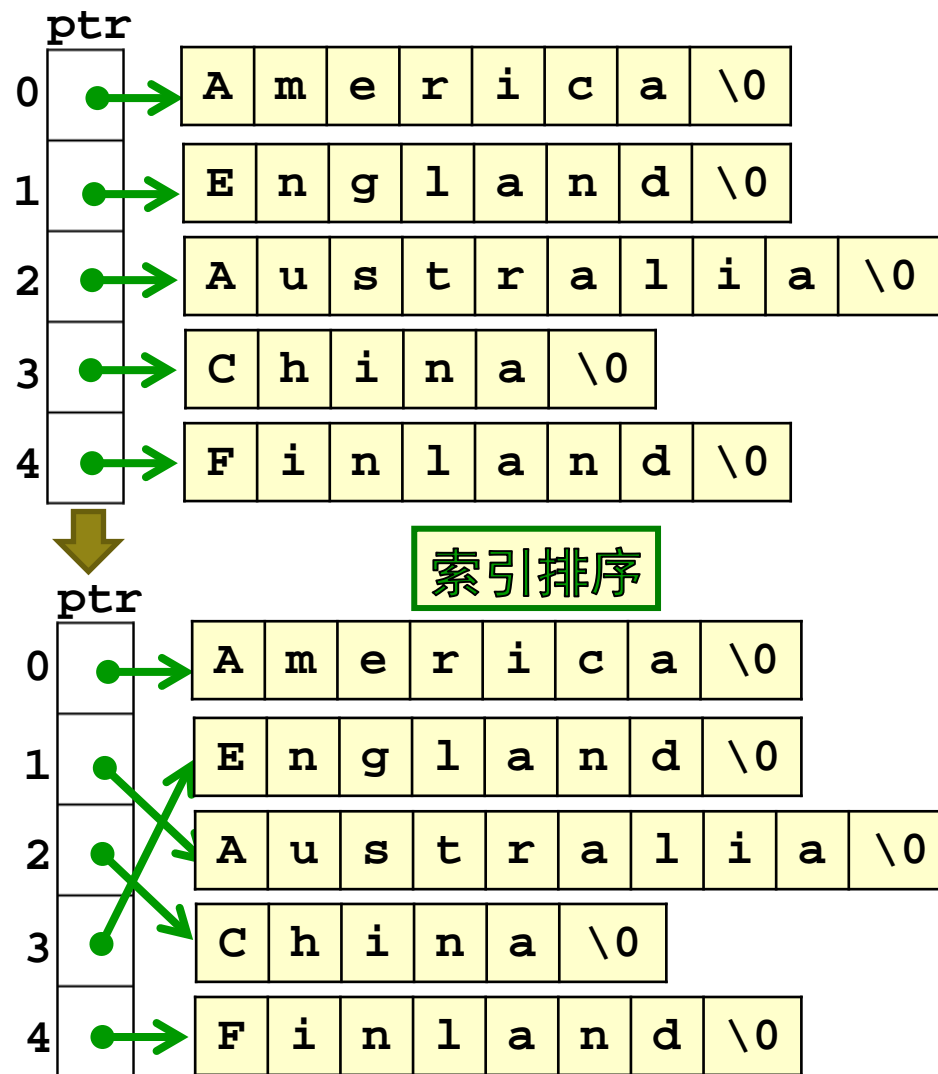
A	m	e	r	i	c	a	\0	\0	\0
A	u	s	t	r	a	l	i	a	\0
C	h	i	n	a	\0	\0	\0	\0	\0
E	n	g	l	a	n	d	\0	\0	\0
F	i	n	l	a	n	d	\0	\0	\0

字符串的排序

按字典顺序对国名排序

```
void SortString(char *ptr[], int n)
{
    int i, j;
    char *temp = NULL;
    for (i=0; i<n-1; i++)
    {
        for (j=i+1; j<n; j++)
        {
            if (strcmp(ptr[j], ptr[i]) < 0)
            {
                temp = ptr[i];
                ptr[i] = ptr[j];
                ptr[j] = temp;
            }
        }
    }
}
```

交换指针数组中的字符串指针



```

#include <stdio.h>
#include <string.h>
#define M 150 /* 最多的字符串个数 */
void SortString(char *ptr[], int n);
int main()
{
    int i, n;
    char *pStr[M];
    printf("How many countries?");
    scanf("%d",&n);
    getchar(); /* 读走输入缓冲区中的回车符 */
    printf("Input their names:\n");
    for (i=0; i<n; i++)
    {
        gets(pStr[i]); /* 输入n个字符串 */
    }
    SortString(pStr, n); /* 字符串按字典顺序排序 */
    printf("Sorted results:\n");
    for (i=0; i<n; i++)
    {
        puts(pStr[i]); /* 输出排序后的n个字符串 */
    }
    return 0;
}

```

讨论

下面主函数调用函数**SortString()**按奥运会参赛国国名在字典中的顺序对其入场次序进行排序，请问程序错在哪里？

How many countries? 5 ✓

Input their names:

America ✓

England ✓

Australia ✓

Sweden ✓

Finland ✓

Sorted results:

America

Australia

England

Finland

Sweden

