اصول ساختمان داده ها در c++

ساختمان داده ها شاخه ای از مهندسی نرم افزار است که به مطالعه انواع ساختارهای داده و نیز الگوریتم های مربوط به آنها می پردازد.

گلزاری پور

آموزشکده فنی و حرفه ای شهید بهشتی کرج

فصل دوم آرایه ها

نوع داده انتزاعی(ADT)

یک مدل ریاضی است که عملیات بر روی آن مدل تعریف شده اند هر نوع داده تشکیل از چند مقدار و مجموعه ای از عملیات بر روی آنها است . مجموعه مقادیر و عملیات یک نوع داده یک ساختار ریاضی را تشکیل می دهد که ممکن است به کمک یک ساختمان داده سخت افزاری یا نرم افزاری پیاده سازی شود.

آرایه

آرایه ساده ترین ساختار داده ای در زبان ++C است و آرا یه یک بعدی مجموعه مرتب و محدودی از عناصر هم گن است.

[1, 2, 3, 4, 5]

انواع ساختار داده

- یک بعدی (نمودار)
- دو بعدی (ماتریس)
- چند بعدی

اعلان آرایه در ++C :

نوع Name [بعد];

مثال: int A [10];

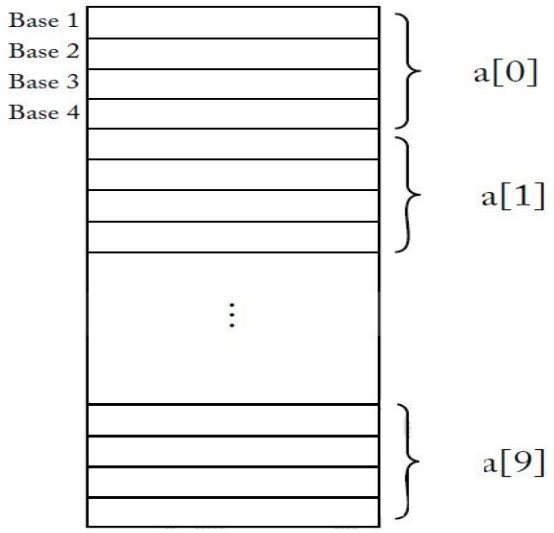
نوع Name [capacity بعد]={ مقدار اولیه } ;

مثال : int B [10] ={1,2,3,4,5,6,7,8,9,10};

اندیس	0	1	2	3	4	5	6	7	8	9
B	1	2	3	4	5	6	7	8	9	10

[1] Abstract Data Type (ADT)

```
int a[ 10 ] ;
```



مثال : فرض کنید آرایه x به صورت int x[10] تعریف شده باشد و base (x) =100 باشد اگر مقدار صحیح چهار بایت را از حافظه اشغال کند آدرس x[3] چند است

Base (x) + I * size

= 100+3*4

=100+12=112

جستوجو در آرایه خطی و دودوئی

جستوجوی خطی: در روش جستوجوی خطی آرایه را از ابتدا تا رسیدن به عنصر مورد نظر (کلید جستوجو) جستوجو میکنیم . به عبارت دیگر ابتدا مقدار key را با A_0 مقایسه میکنیم تا انتها .

مثال :

A	a	b	c	d	e	f	g	h

در صورتی که $key = A_0$ باشد آنگاه عنصر مورد نظر پیدا شده است . در غیر این صورت مقدار key را با A_1 باشد آنگاه عنصر مورد جستوجو پیدا شده است در غیر این صورت key را با $A_1,A_2,...,A_n$ مقایسه میکنیم تا به مقدار مورد جستوجو برسیم.

نحوه نوشتن برنامه جستوجوی خطی

جستوجوی ترتیبی در آرایه های به کار می رود که مرتب نیستند یعنی نه صعودی یا نه نزولی.

```
int search (int a[i] , const int n , int key )
```

```
{
for ( int i=0 ; i<n ; i++ )
if ( a [i] == key )
return i ;
return -1 ;
}
```

بهترین حالت o(1) حالت میانگین o(n/2) بدترین حالت o(n)

جستوجوی دودوئی

جستوجوی دودوئی تنها برای آرایه های مرتب قابل استفاده است . فرض کنید $a=\{a_0,a_1,...,a_n\}$ آرایه ای به طول n باشد آرایه ای از مرتب می گوئیم که در این آرایه را مثل $a=\{a_0,a_1,...,a_n\}$ مرتب می کنیم $a_0 \le a_1 \le ... \le a_{n-}$

نکته

الگوریتم جستوجوی خطی مبنای بسیاری از اعمال رایج برروی آرایه است به عنوان مثال در اعمالی مانند کپی برداری از آرایه و بررسی تساوی در آرایه از روشی مشابه جستوجوی خطی استفاده می شود.

فرایند جستوجو دودوئی به شرح زیر می باشد.

۱- عنصر وسط را پیدا کند.

۲- کلید جستوجو را با عنصر وسط لیست مقایسه کند (سه حالت زیر پیش می آید)

a. مقدار جستو جو با عنصر وسط برابر است . در این صورت عنصر مورد نظر پیدا شده والگوریتم خاتمه می یابد.

b. مقدار جستوجو بزرگتر از عنصر وسط است . در این صورت با توجه به ترتیب صعودی عناصر لیست می توان عمل جستوجو را فقط برای نیمه راست ادامه داد .

c. مقدار جستوجو کوچکتر از عنصر وسط است . در این حالت می توان عکل جستوجو را فقط برای نیمه چپ ادامه داد.

۳- همین روش را برای نیمه چپ یا راست تکرار کنید.

الگوریتم برنامه

```
int Binsearch ( int a[] , cons int n , int key )
{
int mid;
left = 0; right = n-1 ;
while ( left <= right )
        {
mid = ( left + right ) / 2 ;
if ( key == a[ mid ] ) return mid ;
else if ( key < a[ mid ] ) right = mid-1 ;
else left = mid+1 ;
        }
return -1 ; }
```

بهترین حالت o(1) بدترین حالت $o(\log_2 n)$

یافتن بزرگترین و کوچکترین عنصر آرایه :

روش کار با این روش : برای یافتن کوچکترین ابتدا اولین عنصر آرایه را به عنوان کوچکترین عنصر در نظر می گیریم و آن را در متغیری مانندminذخیره میکنیم سپس عناصر بعدی را با متغیرminمقایسه میکنیم هر عنصری که کوپکتر ازminباشد آنرا در متغیرminجایگزین میکنیم واضح است که در پایان کوچکترین عنصر در متغیر min الگوریتم برنامه به شرح زیر می باشد.

```
int findmin ( int a [] , cons tint n )
        {
int min = a[0]
for ( int i=1 ; i<n ; i++ )
if ( a [i] < min )
min = a [i] ;
return min ;
```

```
{
```

درج و حذف عناصر آرایه :

فرض کنید آرایه ای به طول k عنصر باشد و شامل n عنصر معتبر $\{ a_0 , a_1 , a_2 , ... , a_n \}$ باشد

برای انجام عملیات درج و حذف در این آرایه باید نکات زیر را در نظر بگیریم.

۱- شرط پر بودن آرایه این است که n = k باشد که در ابن صورت نمی توان عمل درج دیگری در آن انجام داد

۲- شرط خالی بودن آرایه این است که n= 0 باشد که در این صورت عمل حذف امکان پذیر نیست.

۳- برای درج یک عنصردر مکان pos از آرایه در صورت پر نبودن آرایه باید تمام عناصری که در مکان pos تا n-1 قرار دارد یک مکان به راست انتقال یابد ، سپس عنصر جدید در مکان pos از آرایه قرار گیرد . در این عمل ، اندیس تمام عناصری که در مکان pos تا n-1 قرار دارند ، یکی اضافه می شود . همچنین یک واحد به متغییر n اضافه می شود.

۴- برای حذف عنصری از آرایه که در مکان pos+ ۱ تا k-1 قرار دارد یک مکان به چپ انتقال یابد در این عمل اندیس تمام عناصری که در مکان pos+1 تا k-1 قرار دارند کم می شود. همچنین یک واحد از متغیر n کم می شود.

برای انجام عملیات درج و حذف در آرایه توابع چهارگانه زیر نیاز است:

۱ تست پر بودن آرایه : این تابع در صورت پر بودن آرایه مقدار ۱ ودر غیر این صورت مقدار ۰ را برمیگرداند

```
int IsFull () ;
{
if ( k == n ) return 1 ;
else return 0 ;
{
```

۲ تست خالی بودن آرایه : این تابع در صورت خالی بودن آرایه مقدار ۱ و در غیر این صورت مقدار ۰ را بر می گرداند

```
int IsEmpty()
{
if ( n == 0 ) return 1;
```

```
else return 0 ;
}
```

۳ درج عنصر در آرایه : این تابع عنصر x را در موقعیت pos از آرایه درج می کند.

```
void Insert ( int pos , int x )
{
if ( IsFull() ) cuot << " Array is full. " ;
        else if ( pos > k-1 || pos< 0 )
return ; // InValid pos
else if ( pos > n-1 ) { a [n] = x ; n++ }
else
{
for ( int i = n-1 ; i >= pos ; i -- )
a [i+1] = a [i] ; // Shift Right
a [pos] = x ; // Insert x
} // End else
}
```

۴ حذف عنصر از آرایه : این تابع عنصری که در مکان pos از آرایه قرار دارد حذف می کند و عنصر حذفی را در متغیر x قرار می دهد.

```
void delete ( int pos , int &x )
{
if ( IsFull() ) cout << " Array is empty " ;
else if ( pos >= k-1 || pos < 0 ) return ;
else
{
x =a [pos];
for ( int i = pos + 1 ; i <= n-1 ; i ++ )
a [i+1] = a [i] ; // shift left
n-- ;
```

```
    } // end else
}
```

تبدیل آدرس منطقی به آدرس فیزیکی

آرایه یک بعدی

Base(a)=100

A [1] آدرس عنصر: base (a) + i × size

A [2] آدرس عنصر : ۱۰۰ + ۲ × ۲ = ۱۰۴

آرایه دو بعدی

آرایه دو بعدی ساختار داده ای است که عناصر را با دو بعد شناسائی (بعد سطر و بعد ستون) می نماید و به این شکل تعریف می شود .

نوع نام آرایه [m] [n]

Int B [3] [4];

در زبانهای برنامه نویسی عناصر آرایه های دو بعدی و بالاتر به صورت سطری یا به صورت ستونی در حافظه ذخیره می شود در زبان C++ به صورت سطری ذخیره می شود.

A [i][j]: base (a) + i × size × n + j × size

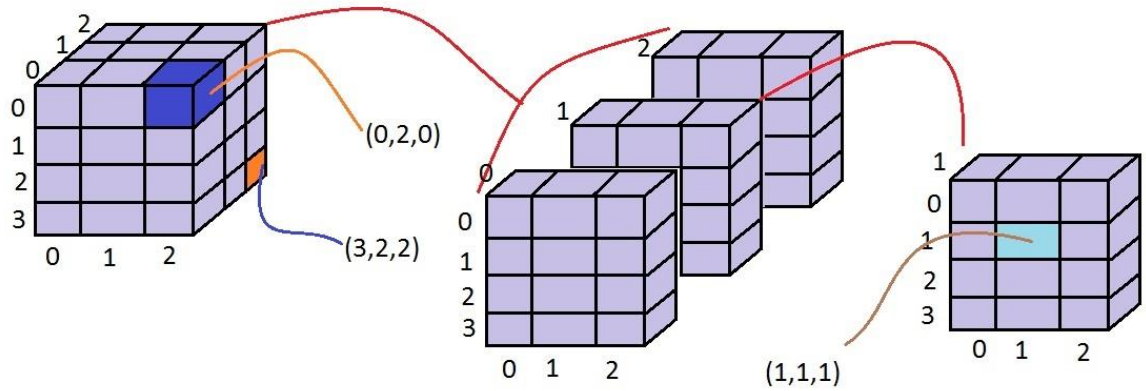
100 + 1× 2 × 2 + 2 ×2 = 108

Base (a) = 100

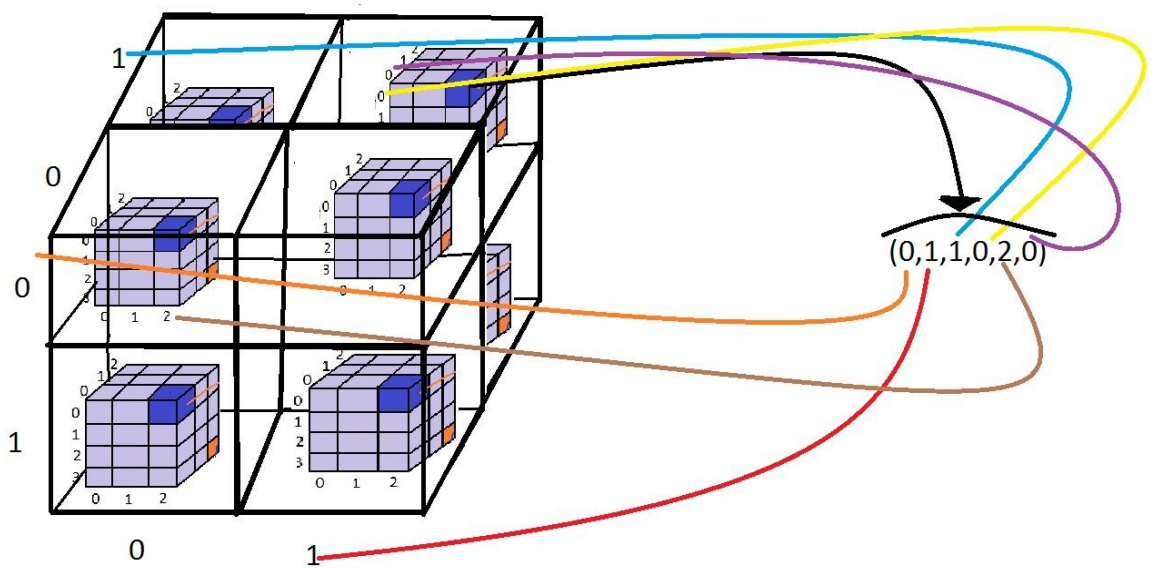
Int size =2B

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & a & b & c \\ 1 & d & e & f \end{array}$$

آرایه ای سه بعدی (۳*۳*٤) (۳٦درایه)



آرایه ای شش بعدی (۳*۳*٤*۲*۲*۲) (۲۸۸ درایه)



آرایه چند بعدی

۲	-۳	۳۳
۴	۱۱	-۵
۶	۴	۱۲
۸	۰	۳

روش ستونی

۲	۴	۶	۸	-۳	۱۱	۴	۰	۳۳	-۵	۱۲	۳

روش سطری

۲	-۳	۳۳	۴	۱۱	-۵	۶	۴	۱۲	۸	۰	۳

ماتریس

آرایه ای دوبعدی است که در شاخه های مختلف علوم ریاضی ، فیزیک ، اقتصاد ، مدیریت و مهندسی کامپیوتر کاربرد دارد. سوال ۱ : جمع دو ماتریس

سوال ۲ ضرب دو ماتریس اگر m*n A و n*p B باشد حاصلضرب آنها ماتریس m*p مانند c میشود الگوریتم ضرب دو ماتریس را بنویسید ؟

ماتریس اسپارس[2]

در بسیاری از کاربرد های علمی مانند حل معاملات خطی معادلات دیفرانسیل و مسائل برنامه ریزی خطی با ماتریس های بزرگی سر و کار داریم که تعداد زیادی ار عناصرشان صفر است اینگونه ماتریس اسپارس یا خلوت می نامند.

برای رسم ماتریس نمایش اسپارس به ابعاد 3 * (n-1) که n تعداد اعداد عناصر غیر صفر است

$$\begin{array}{c c} & \begin{array}{ccccc} 0 & & & 50 & \end{array} \\ \begin{array}{c} 0 \\ \\ 81 \\ \\ 90 \end{array} & \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \dots & & & & \\ 3 & 0 & 0 & \dots & 0 \\ \dots & & & & \\ 0 & 0 & \dots & 2 \dots & 0 \end{bmatrix}_{100*100} \end{array}$$

$$\begin{bmatrix} 100 & 100 & 3 \\ 0 & 0 & 1 \\ 81 & 0 & 3 \\ 90 & 50 & 2 \end{bmatrix}$$

[2] Spars

ماتریس اسپارس (خلوت–Sparse)

$$A=\begin{bmatrix} 5 & 0 & 0 & 20 & 0 & -5 \\ 0 & 12 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 45 & 0 & 0 & 0 \end{bmatrix} \Longrightarrow B=\begin{array}{c} \text{تعداد سطرها} \quad \text{تعداد ستون ها} \quad \text{تعداد عناصر غیر صفر} \\ \begin{bmatrix} 5 & 6 & 7 \\ 1 & 1 & 5 \\ 1 & 4 & 20 \\ 1 & 6 & -5 \\ 2 & 2 & 12 \\ 2 & 3 & 4 \\ 3 & 4 & -8 \\ 5 & 3 & 45 \end{bmatrix} \end{array}$$

دلیل شروع آرایه فوق از ۱ چیست ؟

ماتریس های مثلثی

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{4} & \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{57} & \mathbf{0} & \mathbf{0} \\ \mathbf{5} & \mathbf{0} & \mathbf{8} & \mathbf{8} & \mathbf{0} \\ \mathbf{5} & \mathbf{7} & \mathbf{6} & \mathbf{6} & \mathbf{4} \end{bmatrix}$$

برای نمایش ماتریس های مثلثی دو روش وجود دارد:

نحوه ذخیه سازی ماتریس ها مثلثی (پایین و بالا مثلثی)به دو شکل صورت می گیرد:

استفاده از آرایه دو بعدی

استفاده از آرایه یک بعدی و یافتن فرمولی جهت آدرس دهی

روش اول

از لحاظ مصرف حافظه بهینه نیست زیرا تقریباً نیمی از عناصر ماتریس صفر هستند که نیازی به ذخیره سازی آنها نیست و لی در این روش برای آنها حافظه اختصاص می یابد . در روش اول تعداد کل عناصر برابر است با n^2 (فرض میکنیم ماتریس n*n باشد)

$$\text{حداکثر تعداد عناصر نا صفر} = 1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2} = \frac{3(3+1)}{2} = 6$$

$$\text{حداقل تعداد عناصر صفر} = n^2 - \frac{n(n+1)}{2} = \frac{n(n+1)}{2}$$

روش دوم

استفاده از آرایه یک بعدی و یافتن فرمولی برای آدرس دهی آرایه. (تمرین)

ماتریس مثلثی

ماتریس بالا مثلثی: ماتریسی است مربعی که کلیه عناصر زیر قطر اصلی آن صفر باشد.

$$\begin{bmatrix} 1 & 9 & 0 \\ 0 & 11 & 4 \\ 0 & 0 & -1 \end{bmatrix}$$

ماتریس پایین مثلثی: ماتریسی است مربعی که کلیه عناصر بالای قطر اصلی آن صفر باشد.

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 11 & 0 \\ 6 & 5 & -1 \end{bmatrix}$$

نکته: حداکثر عناصر غیر صفر یک ماتریس مثلثی برابر است با: $n(n-1)/2$

چند جمله ای

$$p(x) = a_n x_n + a_{n-1} x_{n-1} + \ldots + a_1 n_1 + a_0$$

دو روش برای نمایش چند جمله ای ها وجود دارد:

۱- آرایه یک بعدی

۲- آرایه دو بعدی

در روش اول

آرایه یک بعدی به طول n+1 تعریف می کنیم و ضریب هر جمله را از چند جمله ای به شکل x^i

ها در اندیس i ام از این آرایه ذخیره می کنیم

float p [n+1] ;
$p(x) = 5x^4 + x^2 + 2x^0$
float p[5];

2	0	1	0	5
0		2		4

در روش دوم

آرایه دو بعدی شامل دو سطر و n ستون تعریف می کنتیم (n تعداد جملات عنصر صفر است)

سپس در سطر اول توان های جملات عنصر صفر را به ترتیب نزولی توان ها ذخیره می کنیم . پس از آن در سطر دوم ضرایب متناظر با توان های ذخیره شده را ذخیره می نمائیم.

$p(x) = 5x^4 + x^2 +2$
float p[2][3]

توان	۴	۲	۰
ضریب	۵	۱	۲

رشته

دنباله ای متناهی از کاراکتر ها را رشته مینامیم . به عبارت دیگر آرایه ای از کارا کتر ها را رشته گویند .رشته حالت خاصی از آرایه است تعریف و عملیات برای آن نیز مشابه با آرایه انجام می شود در++C انتهای رشته با کاراکتر خاصی بنام '\0'ختم می شود. برای تعریف رشته ای به طول n به صورت زیر عمل می شود

char نام رشته [n];

مثال

رشته ای به طول 8 کاراکتر و با نام str ایجاد و مقدار Iran را در آن ذخیره کنید .

```
char str[8];
char str[8]="Iran";
```

I	r	a	n	\0	?	?	?
0	1	2	3	4	5	6	7

روش های مختلف تطبیق الگو های رشته (الگو های مختلف تطبیق رشته) نام برد و الگوریتم هر یک را بنویسید

s = " abbccbbabcbac"
fat = "abc"

تمرین