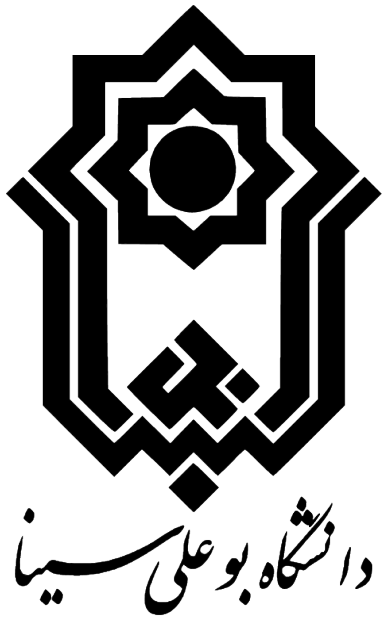


درس برنامه سازی پیشرفته

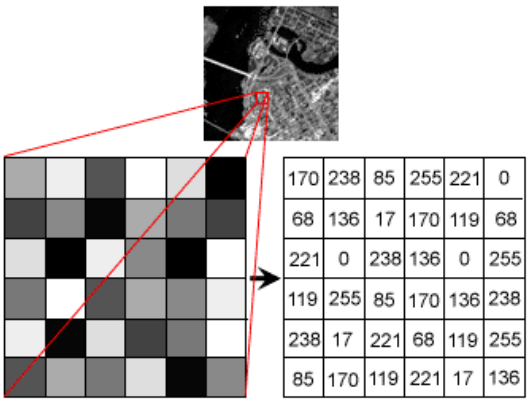
تمرین کار با اشاره گرها

4D & 3D Arrays Implementation

زمستان ۹۷

شرح کلی

در سیستم های کامپیوتری تصاویر دیجیتالی در قالب ماتریس ذخیره می شوند. هر تصویر از تعدادی نقطه یا پیکسل تشکیل شده است. برای مثال یک تصویر با طول و عرض $512px$ دارای 2^{18} پیکسل است. هر درایه‌ی این ماتریس نشان دهنده‌ی یک پیکسل است. در مورد تصاویر سیاه و سفید به هر پیکسل مقداری بین 0 تا 255 پیدا می کند به طوری که عدد 0 نشانگر رنگ سیاه و عدد 255 نشان دهنده‌ی رنگ سفید است. برای مثال تصویر زیر را ببینید.



با این اوصاف واضح است می توانیم یک تصویر سیاه و سفید را در یک ماتریس یا آرایه‌ی دو بعدی ذخیره کنیم.در مورد تصاویر رنگی کلیت کار به همین صورت است. با این تفاوت که نمی توانیم رنگ هر پیکسل را با یک عدد مشخص کنیم. برای اینکه بتوان تمام رنگ های موجود را نمایش داد هر رنگ از ترکیب قرمز، سبز و آبی ساخته می شود. در این نمایش رنگ هر پیکسل با سه عدد که هر کدام بین 0 تا 255 هستند مشخص می شود.برای مثال اگر مقدار این سه رنگ برای یک پیکسل به ترتیب 180 و 70 و 200 باشد آنگاه رنگ این پیکسل بنفش است. از آنجایی که حالا هر خانه از ماتریس دارای سه مولفه است، نمی توانیم این تصاویر را با یک ماتریس عادی متشکل از اعداد نشان دهیم. بنابراین یک روش استفاده از یک آرایه‌ی سه بعدی برای نمایش این تصاویر است. پس دیدیم که یک آرایه‌ی سه بعدی قادر به نگهداری یک تصویر رنگی است. حالا فرض کنیم می خواهیم یک فیلم را نمایش دهیم. هر فیلم از توالی تعدادی تصویر تشکیل شده است و به عبارت دیگر فیلم را می توان آرایه ای از تصاویر دانست. بنابراین می توان از یک آرایه‌ی چهار بعدی برای نمایش فیلم استفاده کرد. هدف اول این تمرین پیاده سازی یک آرایه‌ی سه بعدی برای نگهداری این تصاویر رنگی و پیاده سازی برخی اعمال روی این تصاویر است. هدف دوم این تمرین پیاده سازی یک آرایه‌ی چهاربعدی برای نگهداری یک فیلم است.

مراحل انجام تمرین

در پیاده سازی توابع خواسته شده در این تمرین استفاده از عملگر [ ] برای دسترسی به عناصر آرایه مجاز نیست.

ساخت آرایه‌ی سه بعدی تصویر

تابعی به شکل زیر بنویسید که با دریافت طول و عرض تصویر، یک آرایه‌ی سه بعدی برایش رزرو و آن را با اعداد تصادفی مقدار دهی کند و سپس آن را بازگرداند. اعداد تصادفی بایستی در بازه‌ی $[0, 255]$ باشند.

```
short int*** createImage(short int, short int);
```

نمایش تصویر

تابعی بنویسید که با دریافت تصویر وطول و عرض آن، تصویر را نمایش دهد. نوع تصویر در کد زیر **short int*** در نظر گرفته شده است.اکنون ممکن است مقادیر پیکسل ها و یا **آدرس آنها** به اشتباه در این تابع عوض شود. به جای **short int*** نوع داده ای مناسبی درنظر بگیرید که تضمین شود تصویر **به هیچ طریقی** در این تابع تغییر نخواهد کرد.

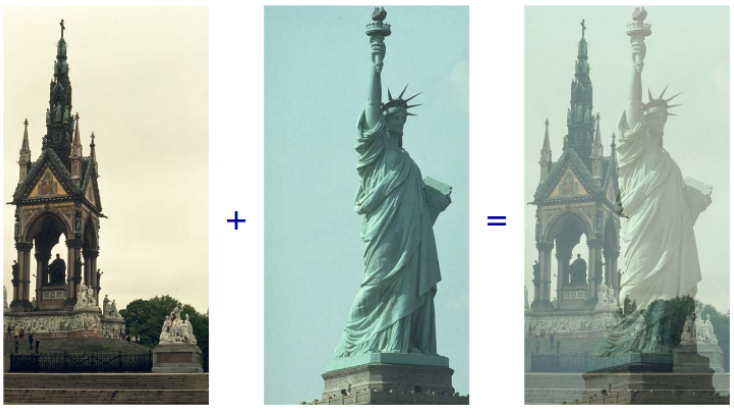
```
void display(short int***, short int, short int);
```

برای نمایش مقدار هر پیکسل، سه مولفه‌ی آن را در [] چاپ کنید.برای مثال خروجی این تابع برای یک تصویر 3×3 می تواند به صورت زیر باشد:

```
[100 255 100] [100 255 100] [255 255 255]
[100 100 100] [210 200 120] [120 100 120]
[100 110 100] [210 150 120] [120 115 120]
```

حاصل جمع دو تصویر

یک روش ترکیب دو تصویر جمع نظیر به نظیر درایه های آن دو است. برای مثال تصویر زیر را ببینید:



دو تصویر را می توان جمع کرد اگر طول و عرض آن ها یکی باشد.تابعی به شکل زیر بنویسید که با دریافت دو تصویر و ابعاد آن ها حاصل جمع آن دو را بازگرداند.

```
short int*** addImage(short int***, short int***, short int, short int);
```

در این تابع نیز **به هیچ طریقی** نباید بتوان دو تصویری که جمع می شوند را تغییر داد.نوع داده ای مناسب را جایگزین کنید.

قدر مطلق اختلاف دو تصویر

با تفریق درایه های نظیر به نظیر دو ماتریس و در نظرگرفتن قدر مطلق آن می توان اختلاف دو تصویر را بدست آورد. برای مثال حاصل تفریق دو تصویر زیر را ببینید.

تابعی به شکل زیر و با همان شرایط عدم تغییر تصاویر ورودی مثل دو تابع قبل برای محاسبه‌ی اختلاف دو تصویر بنویسید:

```
short int*** diffImage(short int***, short int***, short int, short int);
```

چرخش تصویر

تابعی به شکل زیر بنویسید که تصویر را 90 درجه در جهت مثلثاتی دوران دهد. **نوع ورودی ها حتما باید به همین صورت باشد و خود تصویر باید تغییر کند نه اینکه تصویر حاصل را به عنوان خروجی برگرداند.**

```
void rotateImage(const short int***, short int, short int);
```

نگهداری فیلم یا مجموعه ای از تصاویر با آرایه‌ی چهاربعدی

تابعی به شکل زیر بنویسید که آرایه‌ای چهار بعدی برای نگهداری تعداد n تصویر که هر یک به ابعاد $width \times height$ هستند رزرو و آدرس آن را بازگرداند. از اعداد تصادفی در بازه‌ی $[0, 255]$ برای مقدار دهی اولیه تصاویر استفاده کنید.

```
short int**** createVideo(short int n, short int width, short int height);
```

نمایش فیلم ها

تابع display را به شکلی overload کنید که بتواند تصاویر موجود در یک فیلم را به ترتیب نشان دهد.

```
void display(short int ****, short int n, short int width, short int height);
```

نوع `short int****` را با نوع مناسبی جایگزین کنید که فیلم و تصاویر آن **به هیچ طریقی** تغییر نکنند. خروجی این تابع برای یک فیلم حاوی دو تصویر 3×3 می تواند به صورت زیر باشد:

```
Frame 1:
[100 255 100] [100 255 100] [255 255 255]
[100 100 100] [210 200 120] [120 100 120]
[100 110 100] [210 150 120] [120 115 120]
Frame 2:
[100 255 100] [100 50  100] [100 255 255]
[100 40  100] [210 115 120] [200 100 120]
[100 110 100] [210 150 120] [120 115 120]
```

جابجایی دو تصویر در یک فیلم

تابعی به شکل زیر بنویسید که دو تصویر `f1` و `f2` را در فیلم عوض کند. نوع `short int****` را با نوع داده ای مناسبی جایگزین کنید که در بدنه تابع **صرفا جای دو تصویر را بتوان عوض کرد و نتوان خود تصاویر را به هیچ طریقی تغییر داد.**

```
void swapFrames(short int ****, short int f1, short int f2);
```

آزادسازی فضای رزرو شده

تابعی را به صورت زیر overload کنید که فضای رزرو شده برای تصاویر و فیلم ها را آزاد کند.

```
void freeMemory(short int ****, short int n, short int width, short int height);
void freeMemory(short int ***, short int width, short int height);
```

نکات مهم پیاده سازی

- **در پیاده سازی توابع خواسته شده در این تمرین استفاده از عملگر [ ] برای دسترسی به عناصر آرایه مجاز نیست.**
- استفاده از کلاس های vector و array مجاز نیست.
- استفاده از Git اجباری است.

مراجع

در طرح این تمرین از اسلاید های Willy Wriggers در زمینه پردازش تصویر استفاده شده است:

http://biomachina.org/courses/imageproc/032.pdf