



**زیربرنامه:**

EdgeContraction2D

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان** | مرتضی نامور |  |
| امیر همتی زاده | نتیجه تصویری برای دانشگاه تبریز |
| کامیار صفری |  |
| **تهیه کنندگان مستند** | مرتضی نامور، امیر همتی­زاده، کامیار صفری | |
| **تاییدکنندگان** | مرتضی نامور | |
| **تاریخ تنظیم سند** | 4/5/1396 | |
| **شناسه سند** | **MC2F116F1** | |
| **زبان برنامه‌نویسی** | **Fortran 90** | |

1. وظایف

این زیربرنامه عملیات حذف (انقباض) یک ضلع را به طور کامل انجام داده و المانهای مجاور را به روزرسانی کرده و در انتها ضلع مورد نظر را از شبکه حذف میکند.

1. تئوری زیر برنامه­ها

در این زیربرنامه، چندین عملیات که برای حذف یک ضلع ضروری است انجام میشود که در آن از زیربرنامه­های دیگر نیز استفاده خواهد شد. به اختصار میتوان گفت سه عمل بررسی، حذف و بروزرسانی در این زیربرنامه انجام میشود. ابتدا ضلع مورد نظر بررسی شده و IDS به نحوی تنظیم میگردد که همواره نقطه ی مشخص شده ی Dead در محل P1 مربوط به IDS ذخیره شده باشد. اینکار برای جلوگیری از صفر شدن ME یا سلول سمت چپ یک ضلع انجام میگردد. سپس سلولهای چپ و راست و همچنین اضلاعی که به همراه ضلع مورد نظر بایستی حذف شوند و همچنین سلولهای مجاور تشخیص داده میشوند. پس از تشخیص این اطلاعات، سلولهای سمت چپ و سمت راست مربوط به اضلاعی که باقی می مانند، به روزرسانی میگردند. پس از اینکار، عمل انقباض بر روی ضلع قابل حذف انجام میگردد. به این صورت که تمامی اضلاعی که نقطه ی شروع و یا نقطه ی پایان آنها Dead می باشد، آن نقطه برابر با Heir قرار داده میشود. پس از اینکار، بسته به نوع سلولهای سمت چپ و راست و همچنین تغییری که بر روی آن انجام شده است، تصمیم گرفته میشود که یک سلول کلا از شبکه حذف شود یا در شبکه باقی بماند. پس از حذف سلول، اضلاعی که حذف شده اند، کاملا از شبکه و IDS حذف میشوند.

1. بخش‌های زیربرنامه

در این قسمت، توضیح تمامی بخش‌های زیربرنامه، مطابق شماره‌گذاری انجام شده در متن برنامه کامپیوتری ارائه شده است.

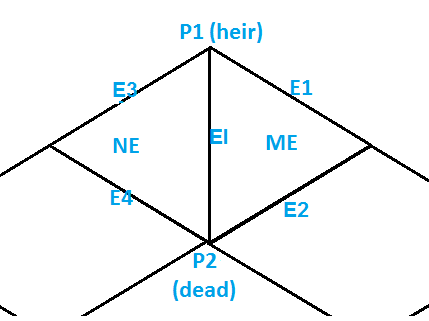
1. صفر کردن متغیرهای مورد استفاده

متغیرهایی که نشان دهنده ی سه ضلعی بودن سلولهای سمت چپ و راست ضلع مورد نظر می باشند، مقداردهی اولیه میشوند. این متغیرها از نوع منطقی می باشند، با مقدار False مقدادهی اولیه می شوند تا نتیجه ی اجراهای قبلی سابروتین بر روی اجرای فعلی اثرگذار نباشد.

1. تعویض جای P1 و P2 در صورتی که P1 برابر Dead نباشد.

در شرایطی ممکن است بعد از انقباض یک ضلع ، یک ضلع مرزی با ME برابر با صفر پدید بیاید که چون اضلاع شبکه به صورت پادساعتگرد ذخیره میشوند، چنین ضلعی باعث بروز خطا در شبکه میگردد.

به عنوان مثال شکل زیر را در نظر بگیرید. فرض کنید میخواهیم ضلع EI را حذف کنیم:



در این حالت محل ذخیره سازی سلول ME در IDS ضلع E2، باید برابر با سلول مجاور ضلع E1 که مخالف ME است قرار بگیرد که در این مثال چون ضلع E1 مرزیست، آن سلول وجود ندارد و شماره ی آن صفر می باشد.

در چنین شرایطی اگر ضلع E2 به نحوی ذخیره شده باشد که ME آن برابر ME ضلع EI باشد،

پس:

1. IDS(1,E2)=IDS(2,E1) => IDS(1,E2)=0

یعنی ME ضلع E2 برابر با صفر میشود و از آنجا که اضلاع به شکل پادساعتگرد ذخیره میشوند، بنابراین ME نمیتواند صفر باشد و اینکار باعث بروز خطا در شبکه میشود.

راه حل این مشکل، این است که در حذف چنین اضلاعی همواره ضلع در IDS به نحوی ذخیره شده باشد که P1 برابر با نقطه ی Dead باشد. در این شرایط در مثال بالا، به جای اضلاع مرزی E1 و یا E3، اضلاع داخلی E2 و یا E4 حذف میشوند و مشکل ذکر شده پیش نمی آید.

قطعه کد این بخش جای نقاط P1 و P2 را با هم و همچنین سلولهای ME و NE را با هم در IDS مربوط به ضلعی که میخواهیم حذف کنیم(EI)، عوض میکند.

1. نگهداری اطلاعات مربوط به ضلع مورد نظر

در این بخش، اطلاعات نقاط شروع و پایان ضلع و همچنین سلولهای سمت چپ و راست ضلع در متغیرهای مربوطه جهت پردازشهای بعدی، نگهداری میشود.

1. یافتن اضلاع و سلولهای مجاور

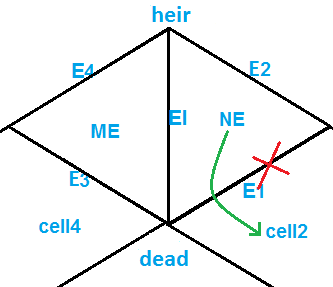
این کار با فراخوانی یک زیربرنامه انجام میشود. این قطعه کد، اضلاع مجاور دو نقطه ی شروع و پایان ضلع قابل حذف، همچنین سلولهای مجاور آنها و همچنین مکان آن سلولها نسبت به اضلاع یافت شده را پیدا میکند. اطلاعات خروجی این سابروتین به این ترتیب می باشد:

* EP1 و EP2: نقاط غیر مشترک متصل به ضلع مورد نظر
* E1 و E2: اضلاع منتهی به نقطه ی EP1
* E3, E4: اضلاع منتهی به نقطه ی EP2
* N1,N3: مکان ذخیره سازی سلولهای مجاور اضلاع E1,E3 در IDS مربوط به آنها
* N2,N4: مکان ذخیره سازی سلولهای ME و NE ضلع EI در IDS اضلاع E2,E4

برای مطالعه ی توضیحات تکمیلی و تئوری این زیربرنامه و نحوه ی سازوکار آن، فایل مستندات مربوط به آن را مطالعه کنید.

1. مقداردهی سلول مشترک ضلع E2 و E1 ضلع E2 برابر با سلول مجاور ضلع E1

کاری که این قطعه کد انجام میدهد، در راستای حذف ضلع E1 می باشد. در شکل زیر فرض کنید ضلع مورد نظر ما برای حذف، EI می باشد. بدیهی است که با حذف EI، با توجه به مثلثی بودن NE، ضلع E1 نیز باید حذف شود. به عبارتی بین ضلع EI و E1 و E2 یک سلول مشترک وجود دارد. در صورتی که این سلول از نوع مثلث باشد، ضلع E1 را حذف و همچنین مکان ذخیره سازی این سلول را در ضلع E2، برابر با سلول غیر مشترک E1، یعنی cell2 قرار میدهیم:



توضیحات شکل:

* E1 حذف شده و NE در IDS مربوط به ضلع E2 برابر Cell2 قرار داده میشود.

نکته1: در صورت مثلث بودن سلول مشترک و انجام تغییرات بالا در شبکه، ، در ادامه ی برنامه آن سلول حذف میشود.

نکته2: در صورتی که سلول مربوطه از نوع مثلث نباشد و چند ضلعی باشد، نیازی به تغییر IDS مربوط به ضلع E2 نیست. زیرا سلول حذف نمیشود و فقط شکل آن تغییر پیدا میکند.

نحوه ی تشخیص اینکه یک سلول مثلث است یا چند ضلعی، همانطور که در این قطعه کد در شرط if آمده است، به این صورت است که ضلع چهارم سلول را بررسی میکنیم. در صورتی که مقدار آن صفر باشد، سلول از نوع مثلث و در صورتی که غیر صفر باشد، سلول چهارضلعی می باشد.

1. مقداردهی سلول مشترک ضلع E4 و E3 ضلع E4 برابر با سلول مجاور ضلع E3

کاری که این قطعه کد انجام میدهد، دقیقا مشابه بخش 5 می باشد. با این تفاوت که این کد برای سلول سمت راست(NE) و اضلاع مربوط به آن اجرا میشود. تفاوت دیگر در این است که باید کنترل کنیم که اگر NE برابر با صفر بود(یعنی ضلع مرزی بود)، قطعه کد اجرا نشود.

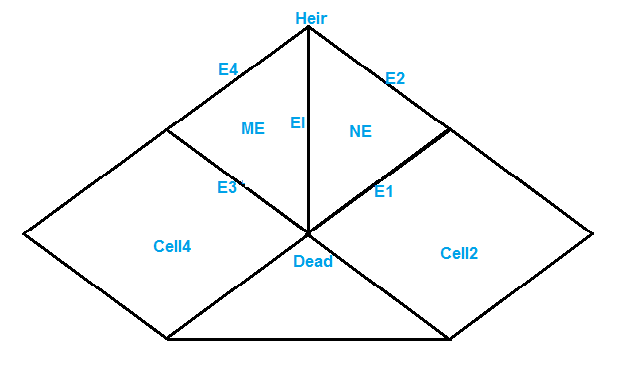
کار کلی این قطعه کد، در راستای حذف ضلع E3 می باشد. به عبارتی ضلع E3 حذف میشود و ME در IDS ضلع E4 برابر با Cell4 قرار میگیرد.

در این بخش نیز، مشابه بخش قبل، در صورتی که سلول مثلثی نباشد(چهار ضلعی باشد)، این قطعه کد نباید اجرا شود.

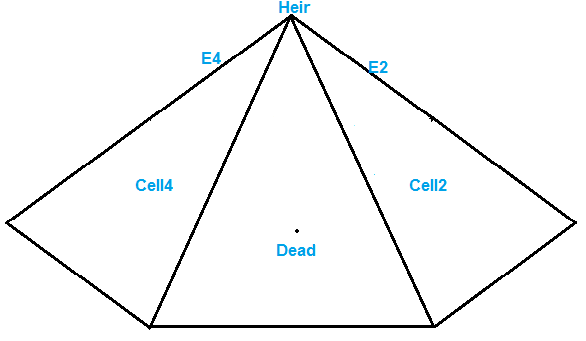
1. تغییر تمامی اضلاع متصل به نقطع ی Dead

در این بخش، تمامی اضلاع متصل به نقطه ی Dead توسط یک حلقه بررسی میشوند. در هر بار پیمایش حلقه، هر دو نفطه ی شروع و پایان یکی از اضلاع متصل به نقطه ی Dead بررسی میگردد. در صورتی که هر کدام از نقاط برابر با Dead باشند، برابر با نقطه ی Dead قرار میگیرند. این کار باعث میشود که عملا ضلع بین Dead و Heir حذف شوند.

در شکل زیر، کار این قطعه کد با یک مثال توضیح داده شده است



پس از انجام اجرای کد و تبدیل تمامی نقاطی که برابر با Dead هستند به Heir :



میبینیم که Dead و تمامی اضلاعی که دارای نقطه ی Dead هستند، به Heir متصل شده اند. به عبارتی ضلع منقبض شده است.

1. شرط ترتیب حذف سلولهای ME و NE مربوط به ضلع EI

به طور کلی قبل از فراخوانی سابروتین حذف سلول، یک شرط اجرا میشود. این شرط باعث میشود که آن سلولی اول حذف شود که شماره ی آن بزرگتر می باشد. دلیل اینکار اینست که اگر سلول کوچکتر در ابتدا حذف شود، چون شماره سلولهای بزرگتر از سلول مورد نظر در IDS تغییر پیدا میکند، در مرحله ی دوم با حذف سلول بزرگتر، سلول اشتباهی حذف نشود.

1. حذف سلولهای چپ و راست(ME و NE) ضلع EI

در این بخش، با فراخوانی یک سابروتین، که شماره ی سلول مورد نظر با به عنوان پارامتر دریافت میکند، سلول مورد نظر را حذف میکنیم. سازوکار کلی حذف یک سلول به این شیوه است که کل اضلاع شبکه پیمایش میشوند. و در IDS شبکه، شماره ی سلولهایی که از سلول حذف شونده بزرگترند، یک واحد کاهش پیدا میکند. به این ترتیب عملا سلول مورد نظر حذف میشود. توضیحات تکمیلی این سابروتین در فایل مستندات مربوط به آن آورده شده است.

قبل از حذف هر سلول، با استفاده از متغیر MEIsTriangle و یا NEIsTriangle، مثلثی بودن یا مثلثی نبودن آن سلول بررسی میگردد و فقط در صورتی سلول حذف میشود که مثلثی باشد. زیرا سلولی که چهارضلعی می باشد، در صورت حذف یکی از اضلاع آن، به یک سلول سه ضلعی تبدیل شده و حذف نمیگردد.

1. حذف سلولهای چپ و راست(ME و NE) ضلع EI

این بخش عینا مشابه بخش شماره 9 می باشد. تنها تفاوت این است که این قسمت زمانی اجرا میشود که شماره ی سلول NE بزرگتر یا مساوی شماره سلول ME باشد. در این صورت ابتدا NE باید حذف شود و پس از آن ME.

1. جلوگیری از حذف اضلاع مجاوری که سلول مشترکشان با EI مثلث نمی‌باشد.

در صورتی که سلول مشترک ضلع مجاور نقطه ی Dead با EI که میتواند ME یا NE مربوط به ضلع EI باشد، یک مثلث باشد، در آن صورت آن ضلع مجاور(E1 یا E3) باید حذف گردد. در غیر این صورت و در صورت چهار ضلعی بودن سلول، باید از حذف آن اضلاع خودداری کرد. جلوگیری از حذف اضلاع E1 و E3 در ادامه ی زیربرنامه، با صفر کردن مقدار این دو ضلع انجام میگردد. پس با استفاده از یک شرط بررسی میکنیم. در صورتی که ME مثلثی نباشد، E1 را برابر صفر و در صورتی که NE مثلثی نباشد، E3 را برابر صفر قرار میدهیم.

1. مقداردهی یک آرایه با استفاده از اضلاعی که باید حذف شوند

برای اینکه بتوانیم به صورت راحتتر لیست سه ضلع EI و E1 و E3 که باید حذف شوند را از بزرگ به کوچک مرتب کنیم، آنها را در یک آرایه ی سه خانه ای به نام Edg میریزیم.

1. مرتبسازی آرایه ی اضلاع EI و E1 و E3 از بزرگ به کوچک

برای جلوگیری از ایجاد اختلال در حذف یک ضلع، لیست اضلاع قابل حذف را از بزرگ به کوچک مرتب سازی میکنیم. اینکار را بدین دلیل انجام میدهیم چون اگر مرتب سازی انجام نگیرد، ممکن است ضلع دوم ضلع آخر کل شبکه باشد، در آنصورت پس از حذف ضلع اول، شماره ی ضلع آخر تغییر پیدا کرده و به جای ضلع دوم ضلعی اشتباهی حذف میشود. روش مرتب سازی، مرتب سازی حبابی(Bubble Sort) می باشد.

1. حذف اضلاع EI و E1 و E3

در این بخش بررسی میشود، هر کدام از اضلاع که مخالف صفر باشند، به طول کامل از ماتریس پیوستگی شبکه یعنی IDS حذف میگردند. نحوه ی حذف یک ضلع به این شیوه است که آخرین ضلع موجود در شبکه به جای ضلعی که قصد حذف آن را داریم در ماتریس پیوستگی IDS قرار میگیرد. سپس تعداد کل اضلاع یک واحد کم میشود. بدین صورت ضلع مورد نظر حذف میگردد. این کد سه بار و هر بار برای یکی از اضلاع EI یا E1 یا E3 که در مرحله ی قبل در آرایه ی Edg ریخته شده اند تکرار میشود و هر بار یکی از این اضلاع را حذف میکند.