فصل ۶

Point Location

Knowing Where You Are

ارائه دهنده:

حسین صفری

پاییز ۹۳

مسأله مکان‌یابی

مسأله مکان‌یابی

هدف ما این است که پیش پردازشی روی نقشه انجام داده و آن را ذخیره کنیم و یک ساختمان داده بسازیم که مکان یابی نقطه ای را سریع انجام دهد.

صورت مسأله

ورودی : یک زیر بخش مسطح S با n یال و یک نقطه q است

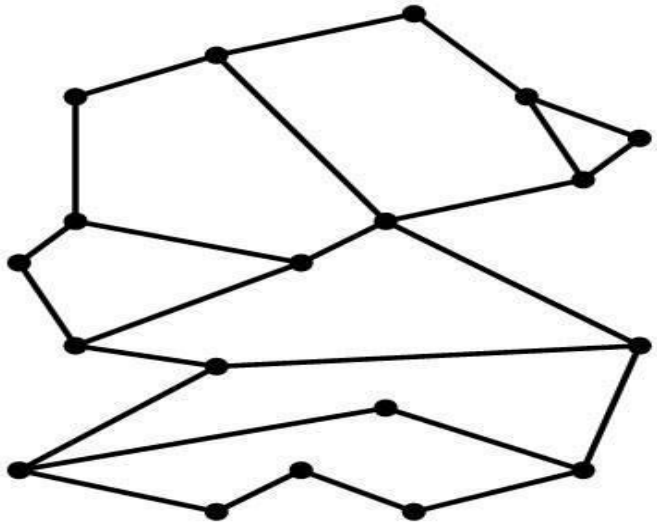
خروجی: تعیین ناحیه ای از زیربخش S است که q درآن قراردارد.

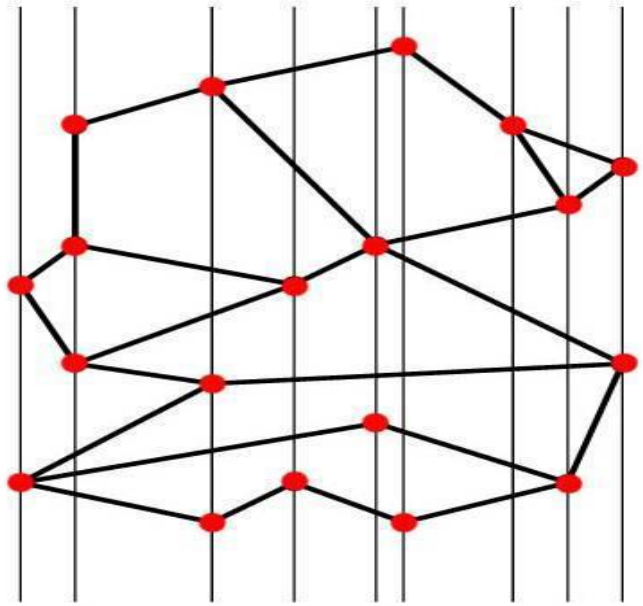
توجه: نقشه یک زیر بخش مسطح S می باشد.

مکان یابی با نقشه ذوزنقه‌ای

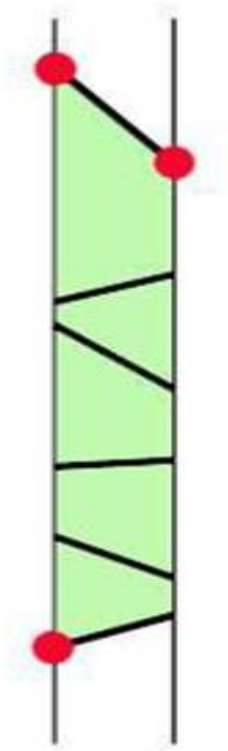
از هر رأس از زیربخش یک خط عمودی رسم می کنیم.

زیر بخش در صفحه به نوارها عمودی افراز می شود .(مختصات X رئوس به ترتیب مرتب شده در یک آرایه ذخیره می کنیم)





مکان یابی با نقشه ذوزنقه‌ای



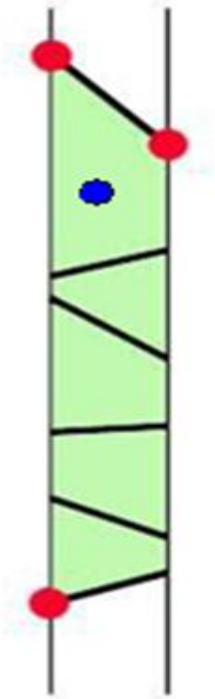
در هر نوار عمودی یال هایی وجود دارند که یکدیگر را قطع نمی کنند.(یعنی یال‌های یک نوار را می توانیم به ترتیب (از بالا به پایین) در یک آرایه مرتب کنیم)

هرناحیه از نوار بین دو یال پشت سر هم قرار دارد که شامل یک ناحیه از زیربخش می باشد.

دو ناحیه بیکران در بالا و پایین قرار دارند.

هر یال با face بلافاصله بالای آن برچسب شده است.

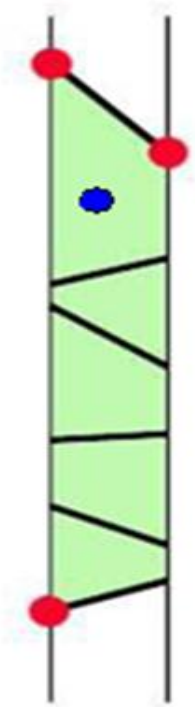
مکان یابی با نقشه ذوزنقه‌ای



نحوه مکان یابی نقطه:

- جستجوی دودویی با مختص x برای تعیین نواری که نقطه در آن است.
- جستجوی دودویی با مختص y برای تعیین پاره خطی که نقطه در بالای آن قرار دارد.

مکان یابی با نقشه ذوزنقه‌ای

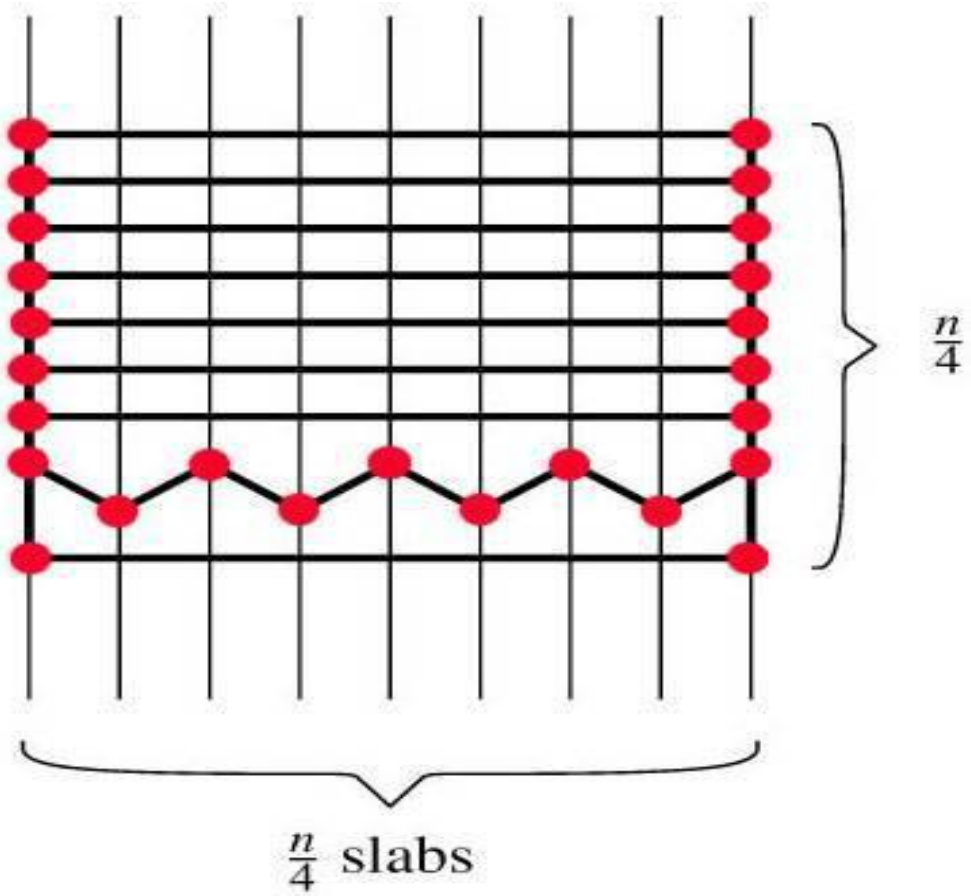


تحلیل زمان پرس‌وجو و حافظه مورد نیاز:

- جستجوی دودویی با مختص x O(log n)
- جستجوی دودویی با مختص y O(log n)
- زمان جستجو :O(log n)
- حافظه لازم: O(n) نوار و هر نوار O(n) پاره خط دارد. $O(n^2)$

نکته: در این الگوریتم زمان جستجوی خوب بوده اما دارای حافظه مصرفی $O(n^2)$ بوده که در اکثر کاربردهای عملی که میانگین n بزرگ باشد بدون استفاده است.

مکان یابی با نقشه ذوزنقه‌ای



برای نمونه این کران بالا ممکن است اتفاق بیفتد.

زیر بخشی با n یال، $n/4$ نوار که هر نوار $n/4$ ناحیه دارند.

در این روش یک زیربخش را به یک زیربخش دیگر تبدیل کردیم که جستجو در آن ساده تر است اما این زیربخش دارای $O(n^2)$ ناحیه(face) است.

مکان یابی با نقشه ذوزنقه‌ای

یک راه حل بهتر:

استفاده از روش نقشه ذوزنقه‌ای برای کم کردن تعداد ناحیه‌ها.

پاره خط‌های غیر متقاطع :

دو پاره خط را غیر متقاطع می نامیم هر گاه تقاطع نداشته باشند و یا در یک رأس انتهایی مشترک متقاطع باشند.

نقشه ذوزنقه ای:

یک مجموه S از n پاره خط غیر متقاطع در صفحه مفروض می‌باشد.

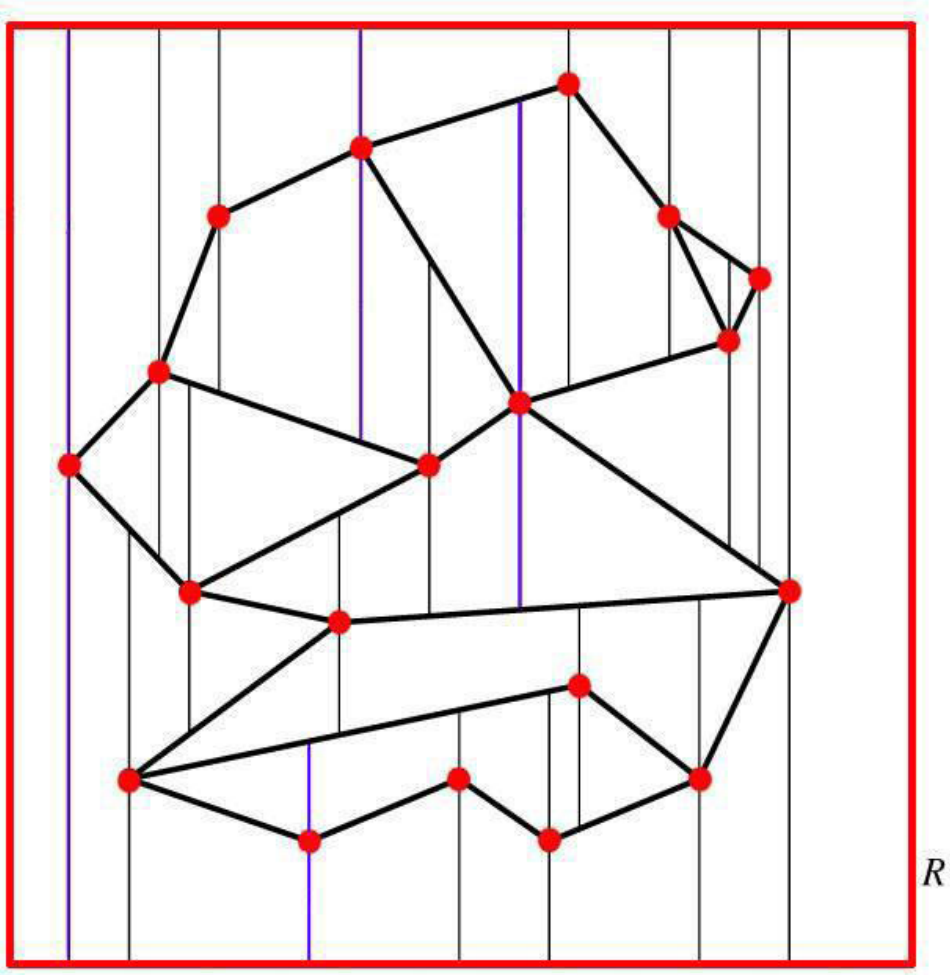
دو ساده سازی برای راحتی کار:

۱. زیربخش را با یک مستطیل R محدود می کنیم.

۲. هیچ دونقطه انتهایی از پاره خط ها دارای مختص x یکسان نیستند.

این پاره خطها در موقعیت کلی نامیده می شوند.

از هر رأس زیربخش دو پاره خط عمودی یکی به طرف بالا و یکی به طرف پایین رسم می شود.تا تا جایی که به یک یال S و یا مستطیل R برسند.

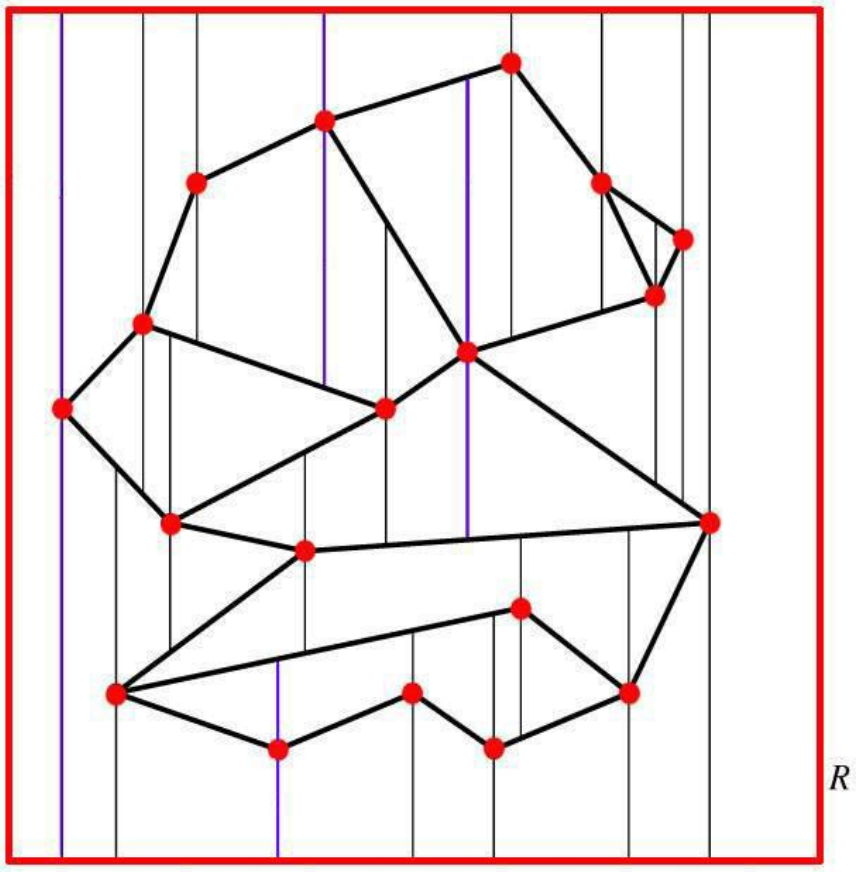


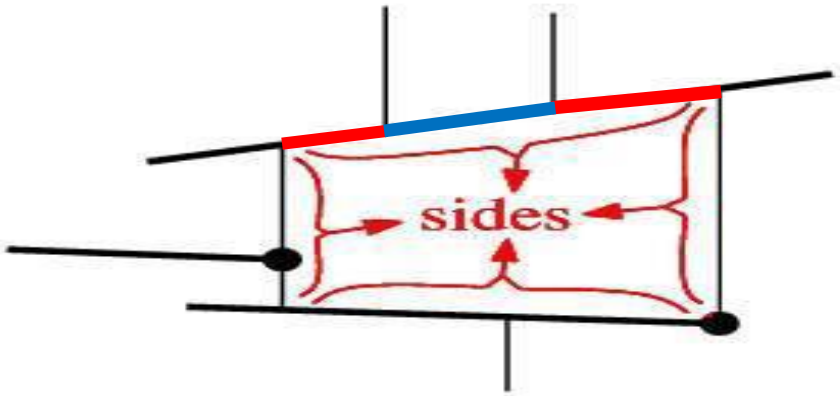
نقشه ذوزنقه ای

زیر بخش بدست آمده را نقشه‌ی ذوزنقه T(S) می نامیم.

یک face در T(S) با تعدادی از یال‌های T(S) محدود شده.ممکن است تعدادی از این پاره خط‌های مجاور و هم راستا باشند.

اجتماع چنین پاره خط هایی را یک وجه (side) می نامیم.

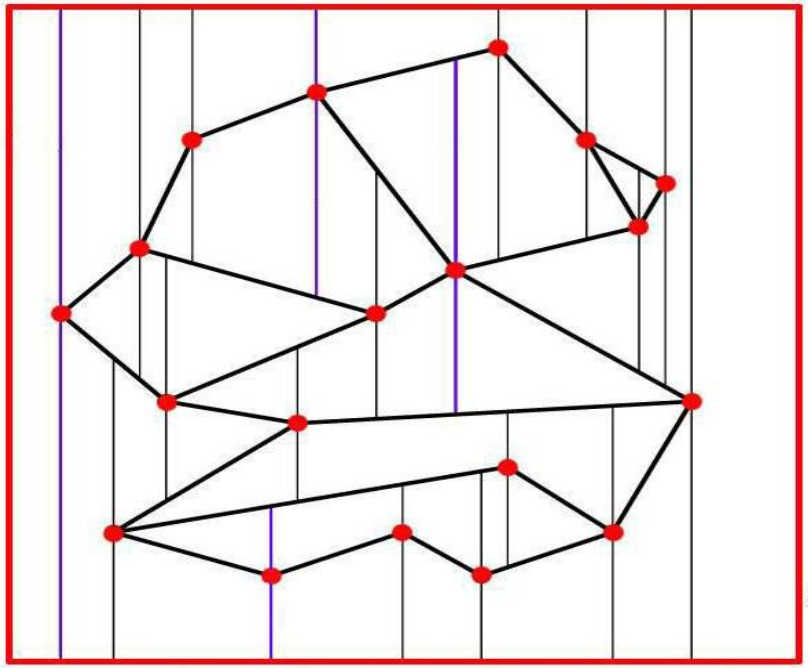




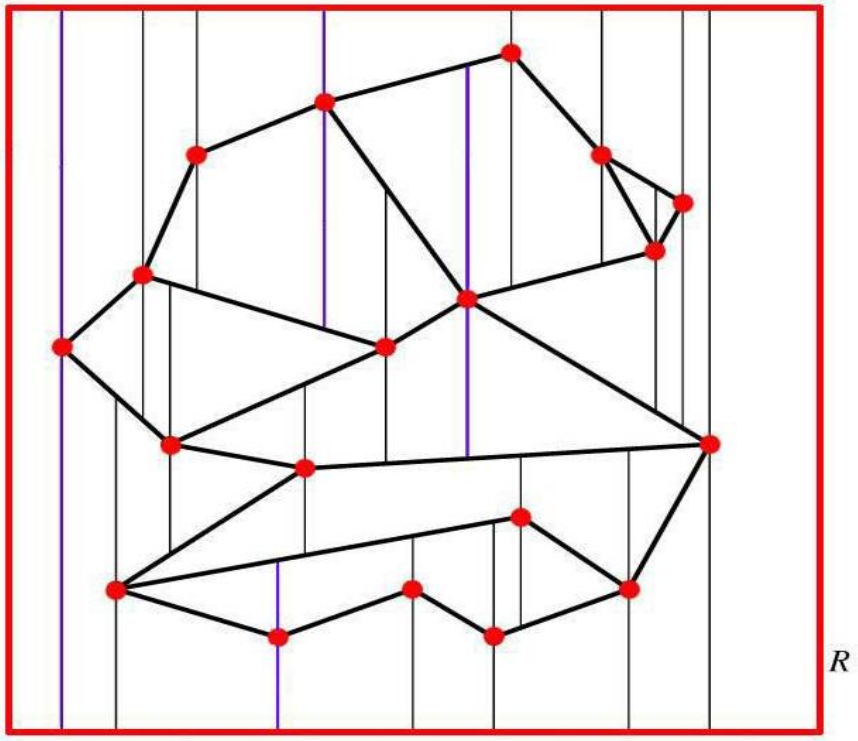
قضیه ۶.۱: هر face در نقشه ذوزنقه‌بندی شده ازیک مجموعه S شامل پاره خط هایی که در موقعیت کلی قرار دارند، یک یا دو وجه عمودی و دقیقا دو کناره غیر عمودی دارد.

اثبات: اگر f وجهی از T(S) با شد، f محدب است زیرا همه زاویه های داخلی کمتر از ۱۸۰ درجه است.

از آنجای که ما به مرزهای وجه‌های f به جای یالهای T(S) نگاه می‌کنیم و محدب بودن f نشان می دهد که f حداکثر دو وجه عمودی دارد. و باید حداقل یک وجه عمودی وجود داشته باشد.

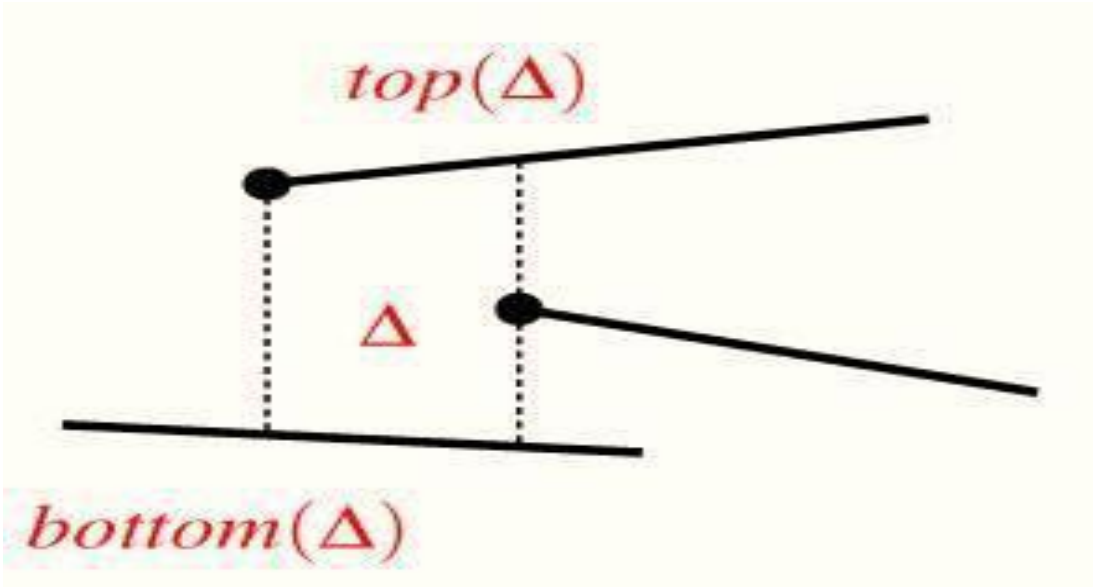


اگرf بیش از دو وجه غیر عمودی داشته باشد دو تای آنها در بالا یا پایین قرار دارند. یال های زیربخش تقاطع ندارند، پس این دو پاره خط دریک رأس انتهایی مشترک به هم می رسند. که در آن یک امتداد عمودی رسم شده است و در نتیجه هر دو پاره خط در یک وجه نیستند.پس f حداکثر دو وجه غیر عمودی دارد. و نهایتا f توسط R کران دار شده، این نشان می دهد که نمی‌توانیم کمتر از دو وجه غیر عمودی داشته باشیم.



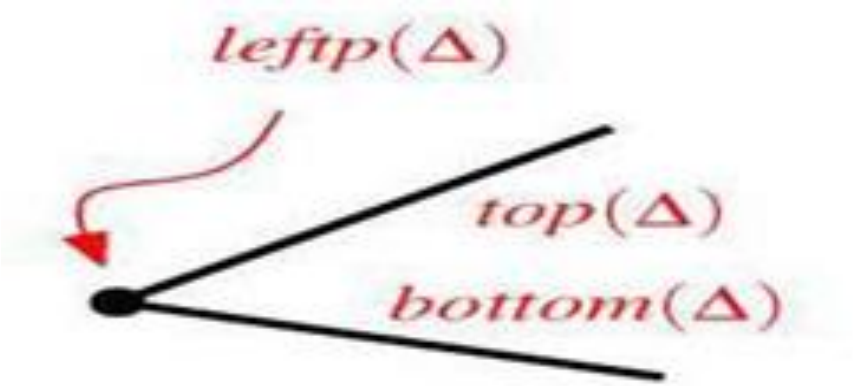
افراز ذوزنقه ای

با توجه به اثبات قضیه ۶.۱ اگر Δ یک face از افراز ذوزنقه ای باشد وجه‌های بالا و پایین آن را به ترتیب $top(\Delta)$ و $buttom(\Delta)$ می نامیم.

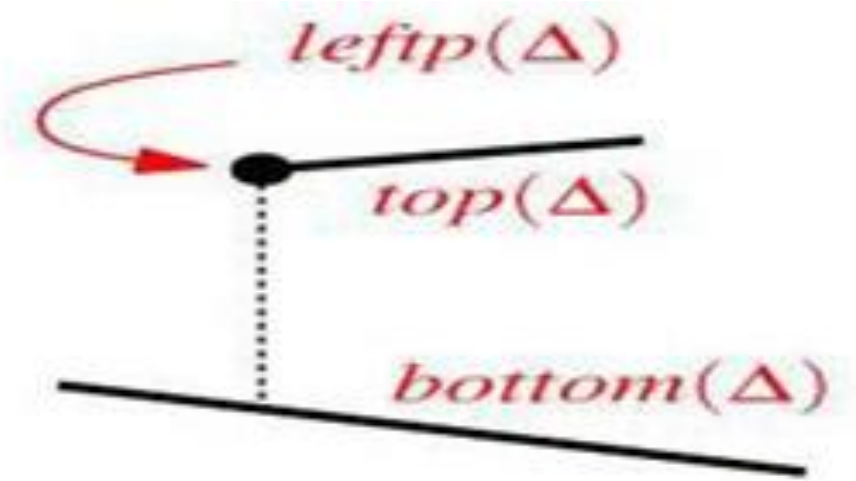


افراز ذوزنقه ای

می‌توان ۵ حالت مختلف برای برای وجه راست و وجه چپ یک ذوزنقه Δ در نظر گرفت.

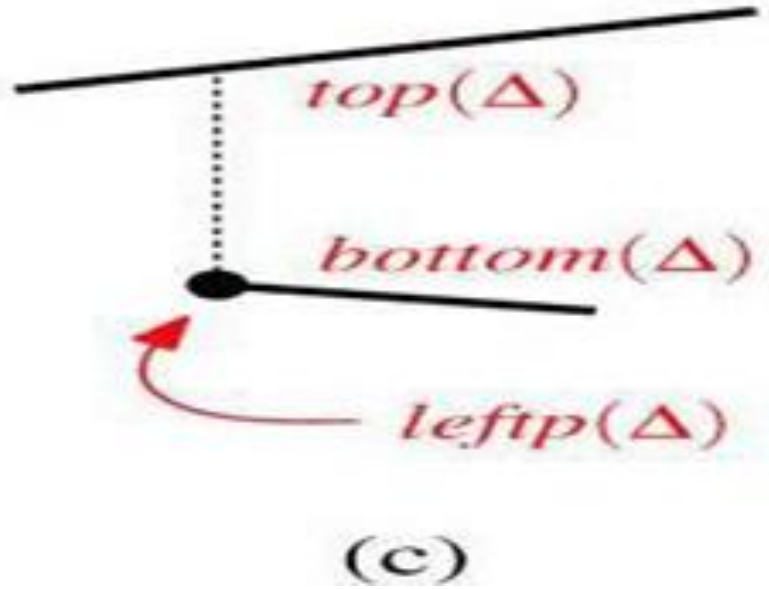


(a)

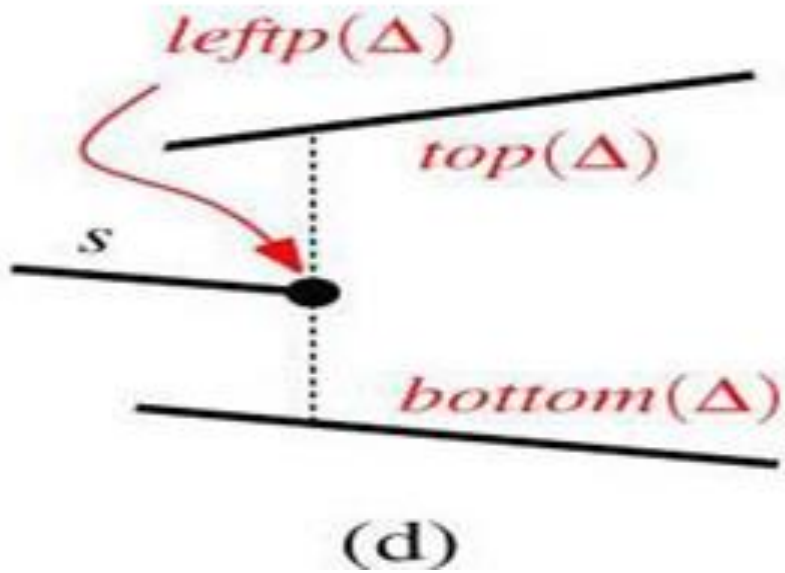


(b)

افراز ذوزنقه ای

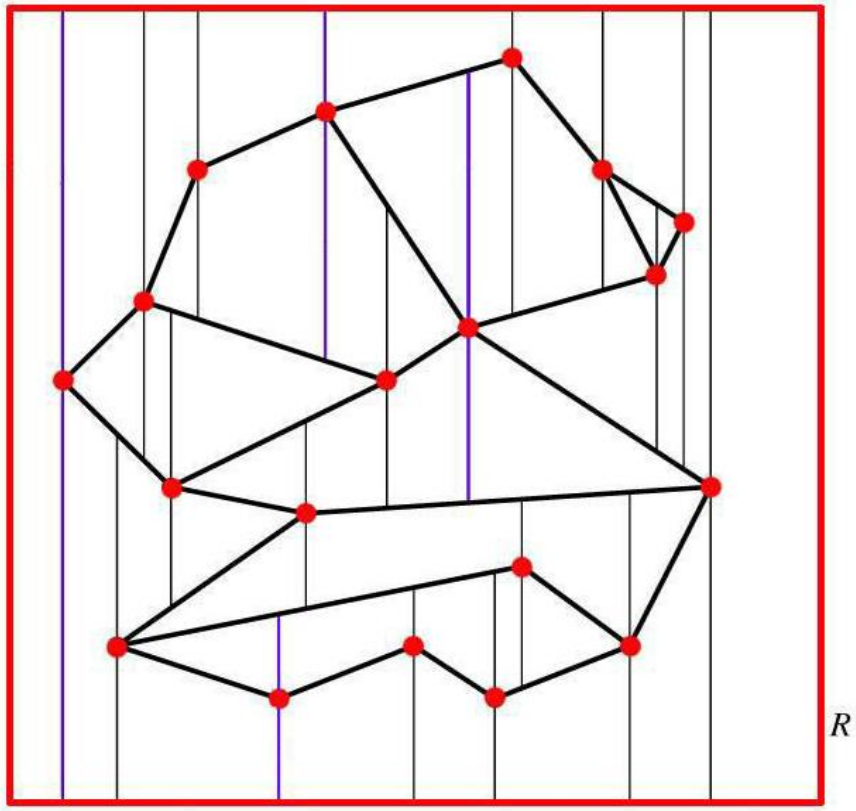


(c)



(d)

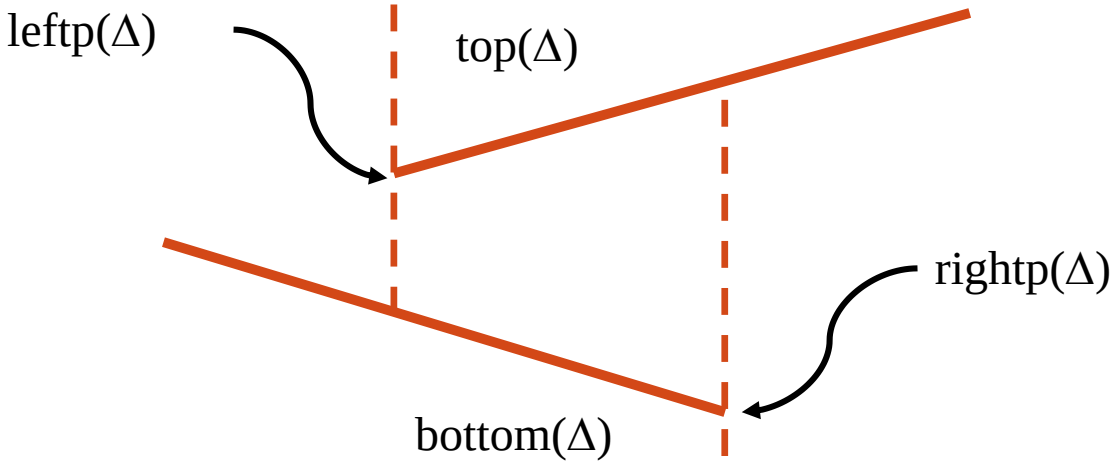
افراز ذوزنقه ای

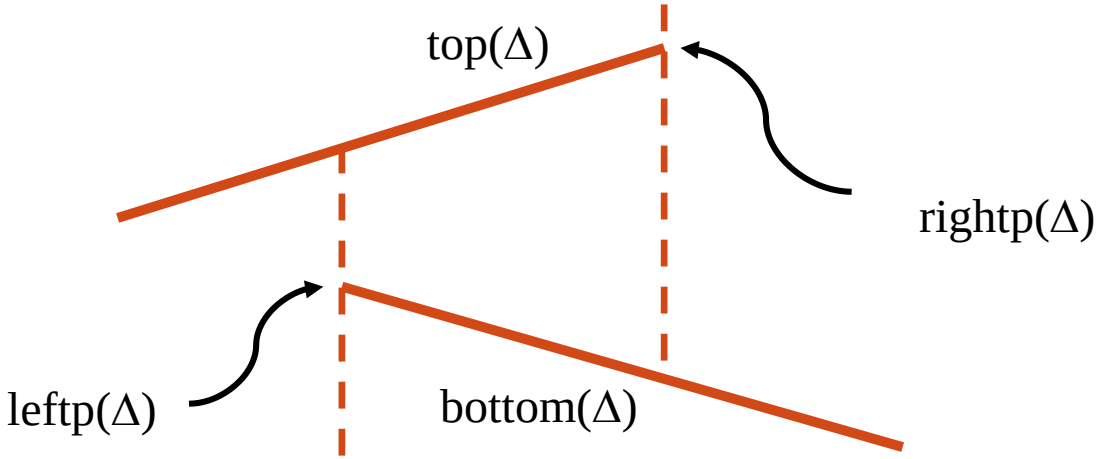


(e)

افراز ذوزنقه ای

هر وجه Δ باداشتن top(Δ)، bottom(Δ)، rightp(Δ)، leftp(Δ) به طور منحصر به فرد تعیین می شود.



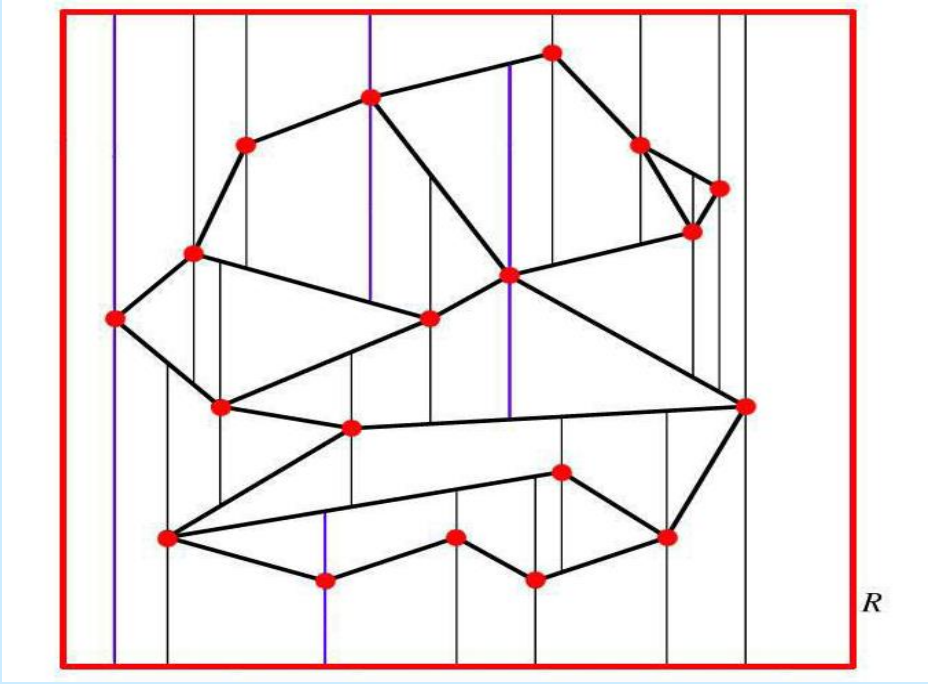


قضیه۶.۲: افراز ذوزنقه ای $T(S)$ از مجموعه S با n پاره خط در موقعیت کلی، حد اکثر $6n+4$ رأس و حداکثر $3n+1$ ذوزنقه دارد.

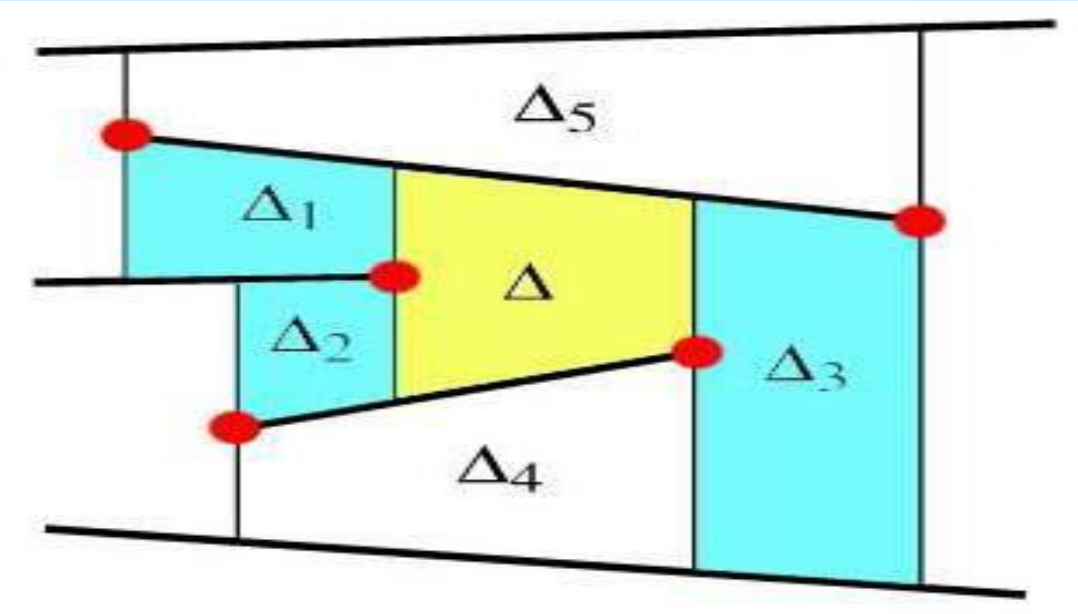
اثبات :

هر رأس افراز ذوزنقه ای رأسی از s، رأسی از R یا رأسی درمحل تقاطع یک گسترش عمودی و یک پاره خط است . پس تعداد رأس ها برابر است با :

$$4+2n+2(2n)=6n+4$$

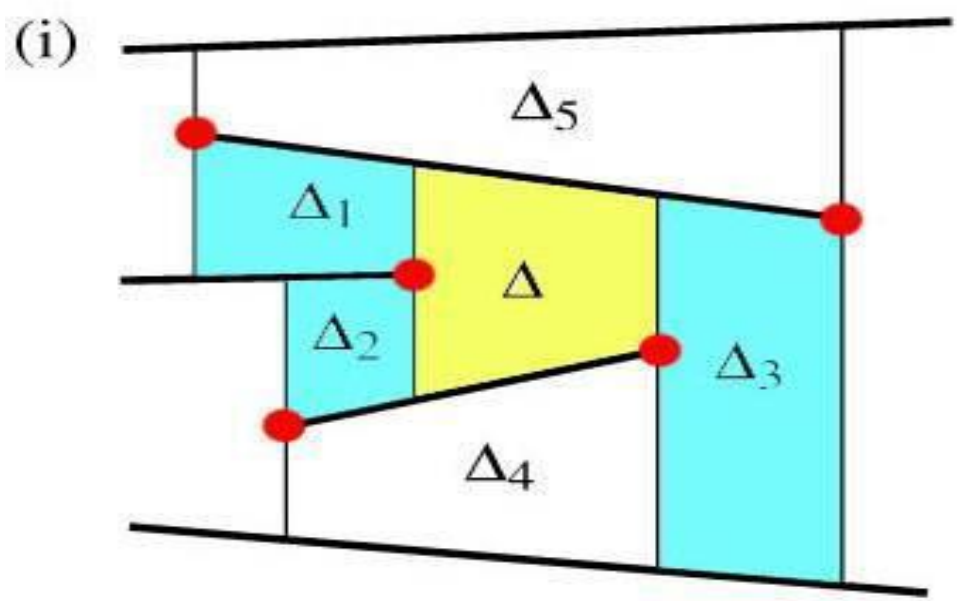


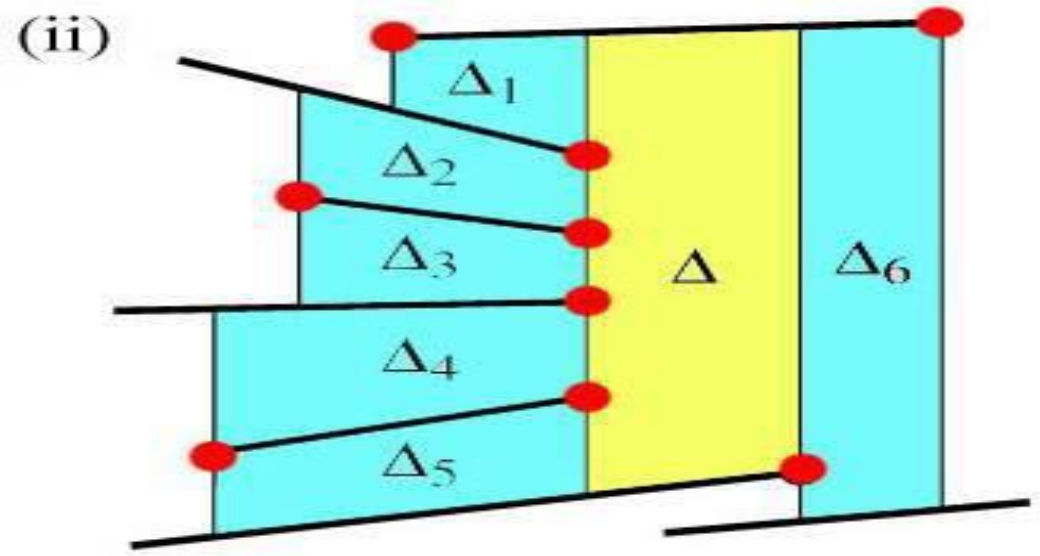
تعداد ذوزنقه‌ها با تعداد رأس های سمت چپ آنها برابر است. رأس انتهای سمت چپ هرپاره خط از S برای حداکثر۲ ذوزنقه و رأس سمت راست آن برای حداکثر یک ذوزنقه به عنوان leftp انتخاب می شود. تنها یک رأس از R به عنوان leftp انتخاب می شود پس تعداد وجه ها از $3n+1$ بیشتر نمی شود.



افراز ذوزنقه ای

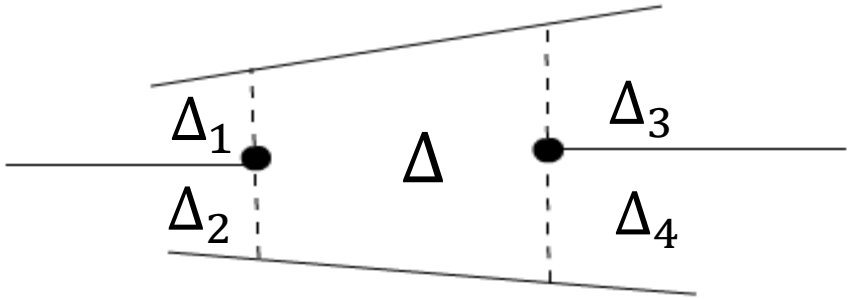
اگر Δ یک وجه از افراز ذوزنقه ای باشد ، وجه Δ' را مجاور با Δ می نامیم هرگاه Δ و Δ' یک یال عمودی مشترک داشته باشند.



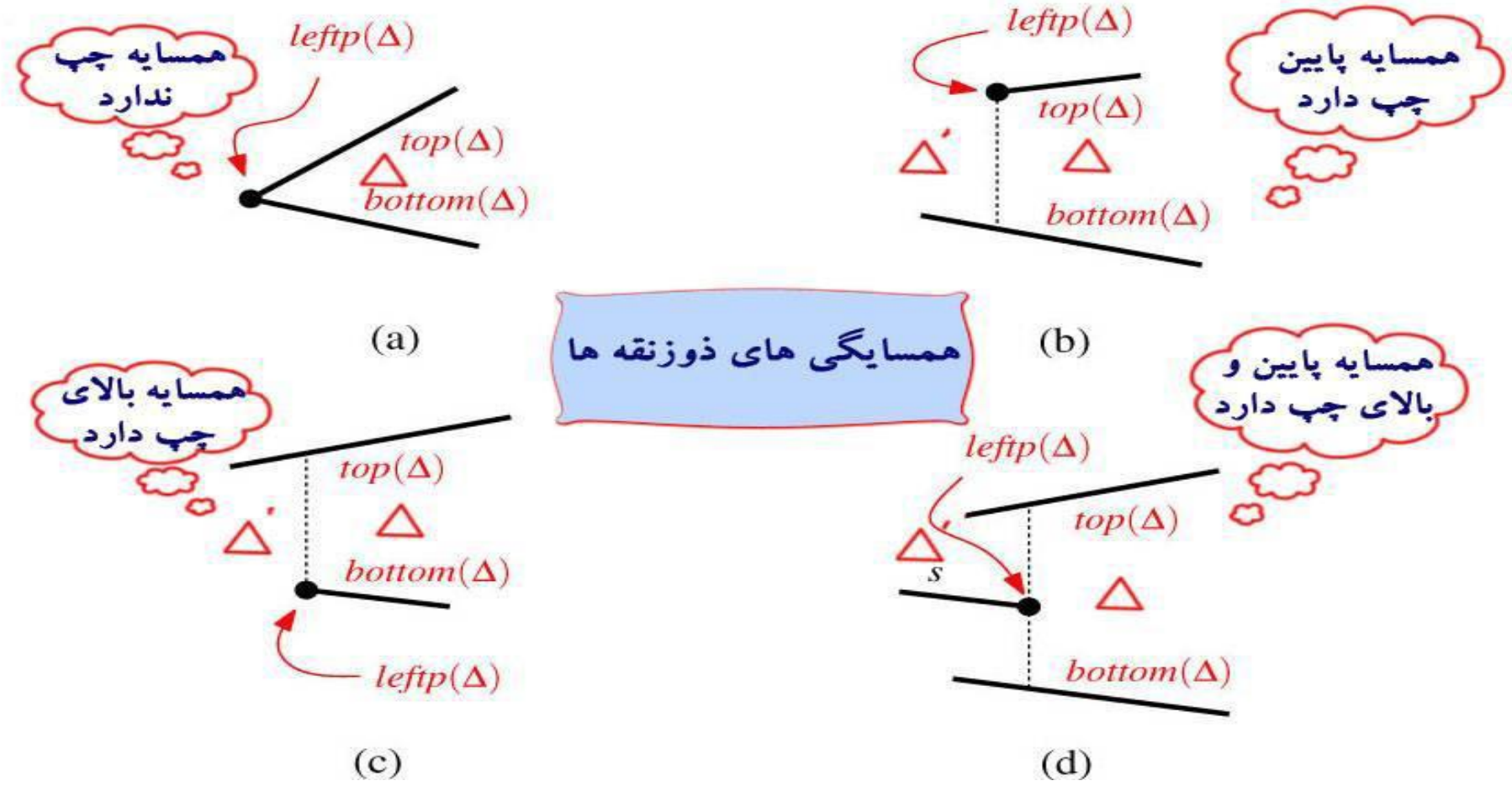


افراز ذوزنقه ای

با توجه به این که پاره خط ها در موقعیت کلی هستند هر ذوزنقه با حداکثر۴ ذوزنقه دیگر همسایه است.



افراز ذوزنقه ای



افراز ذوزنقه ای

یک افراز ذوزنقه ای یک زیر بخش است پس می توان آن را با یک لیست دو همبند یالی ذخیره کرد.

اما با توجه به ساختار خاص نقشه ذوزنقه ما می توانیم با استفاده از یک ساختار خاص راحتر از افراز ذوزنقه‌ای استفاده می کنیم.

با توجه به اینکه هر ذوزنقه Δ به طور منحصر به فردی با top، bottom ،leftp و rightp مشخص می شود ، برای هر ذوزنقه Δ یک لیست متشکل از top، bottom، leftp وrightp و اشاره گرهایی به حداکثر ۴ همسایه آن درنظر می گیریم .

برای این کار یک رکورد برای پاره خطها و یک رکورد برای نقطه‌های انتهایی پاره خطها داریم.

در این ساختار راس‌های ذوزنقه را می توان در زمان ثابت تعیین کرد.

ساختن افراز ذوزنقه ای

الگوریتم تصادفی افزایشی :

ورودی: یک مجموعه S از پاره خط ها که در موقعیت کلی قرار دارند.

خروجی: یک افراز ذوزنفه ای $T(S)$ با ساختمان داده D که برای مکان یابی در افراز ذوزنقه ای به کار می رود.

ساختن افراز ذوزنقه ای

معرفی ساختمان‌داده D:

ساختمان داده D یک گراف جهتدار بدون دور، با یک رأس یکتای ریشه و یک برگ به ازای هر ذوزنقه ازافراز ذوزنقه ای S است.

دو نوع رأس داخلی:

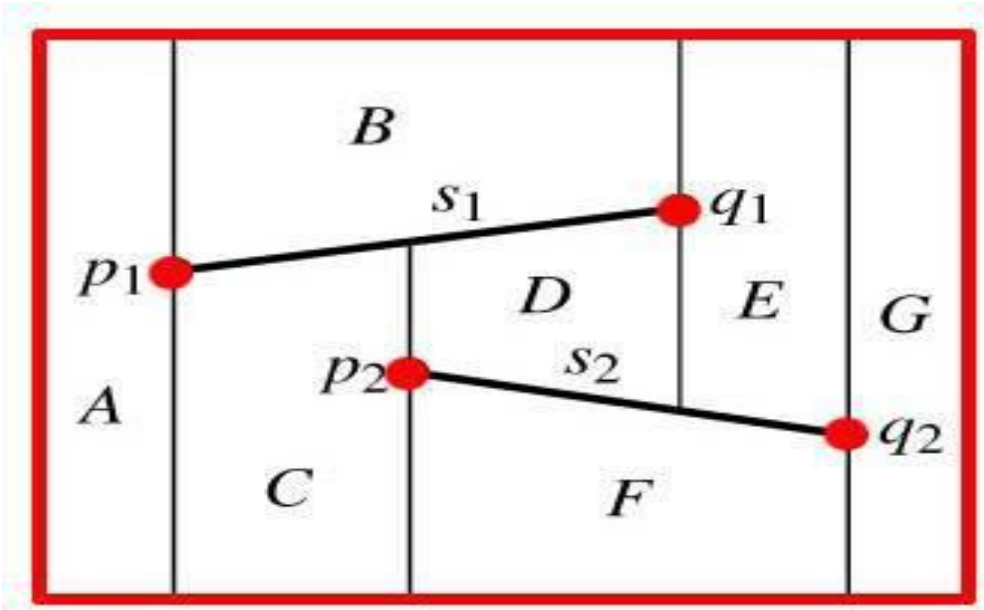
X- رأس ها: با یک نقطه انتهایی پاره خط‌های از S برچسب شده‌اند.

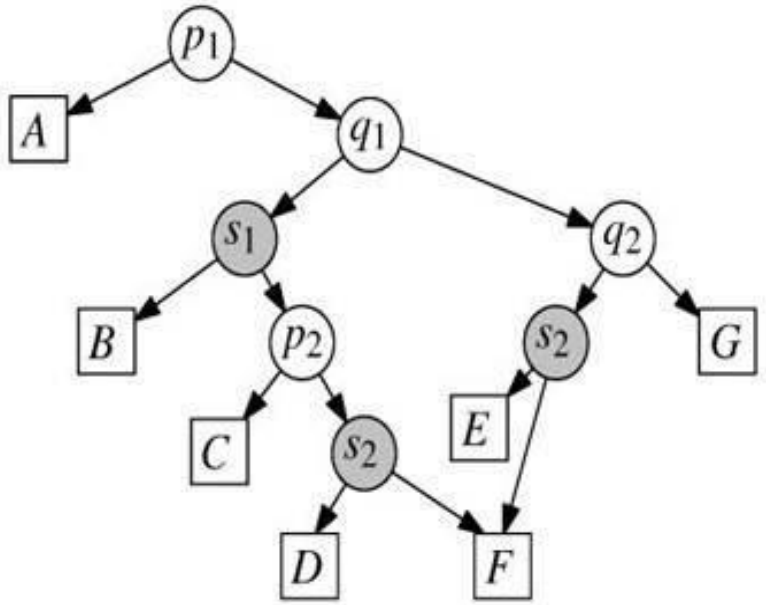
Y- رأس ها : با یک پاره خط برچسب گذاری شده اند.

هررأس داخلی درجه خروجی ۲ دارد.

ساختن افراز ذوزنقه ای

معرفی ساختمان‌داده D:





ساختن افراز ذوزنقه ای

جستجوی یک نقطه در D:

برای جستجوی نقطه q از ریشه شروع میکنیم.

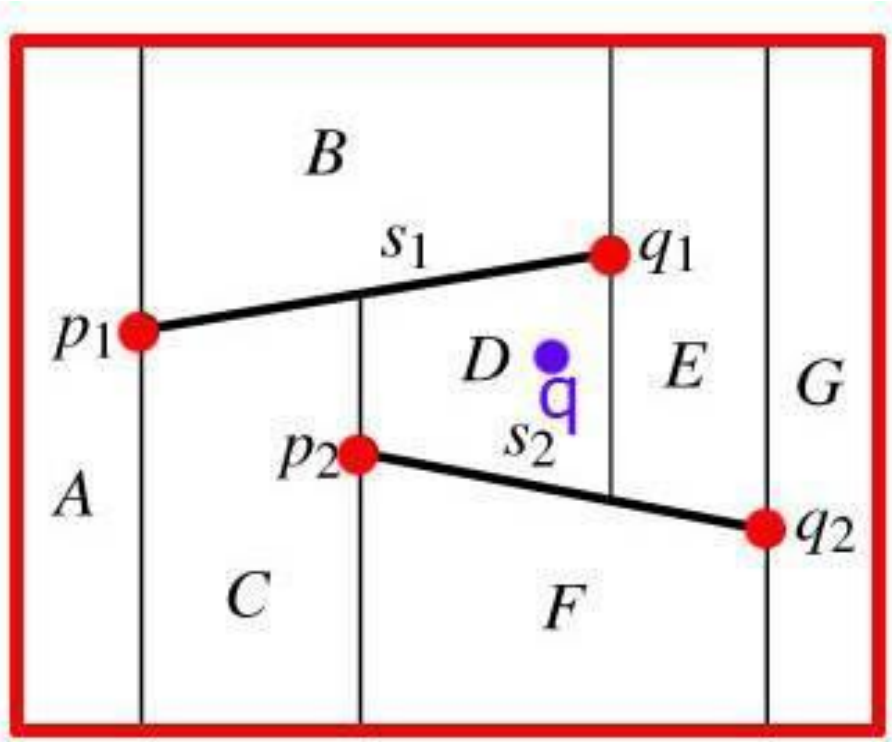
در هر X راس: آیا q سمت راست یا چپ نقطه ذخیره شده در این راس است؟

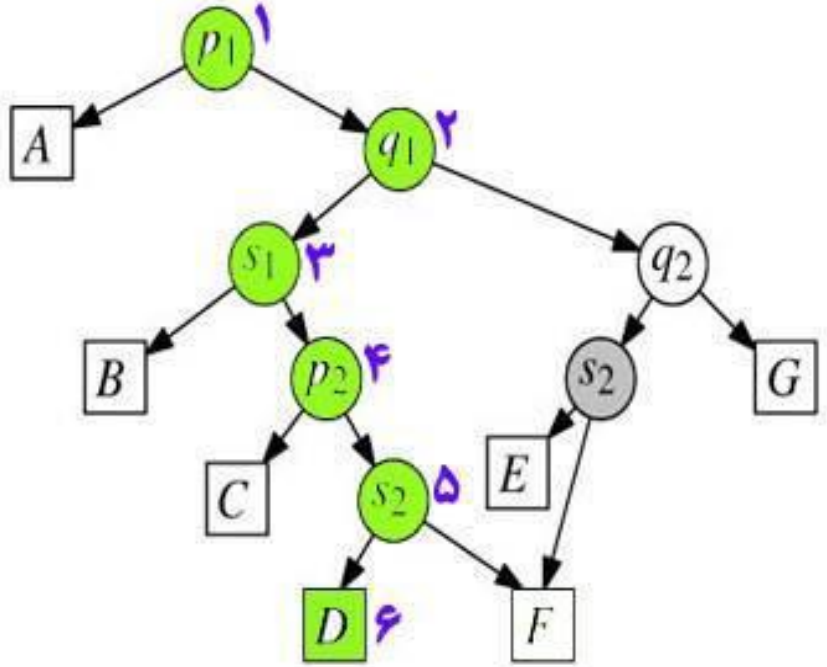
در هر Y راس: آیا q در بالا یا پایین پاره خط ذخیره شده در این راس است؟

جستجو با یک برگ D تمام می شود که ذوزنقه ای که q در آن است را نشان می دهد.

ساختن افراز ذوزنقه ای

جستجوی یک نقطه در D :





الگوریتم:

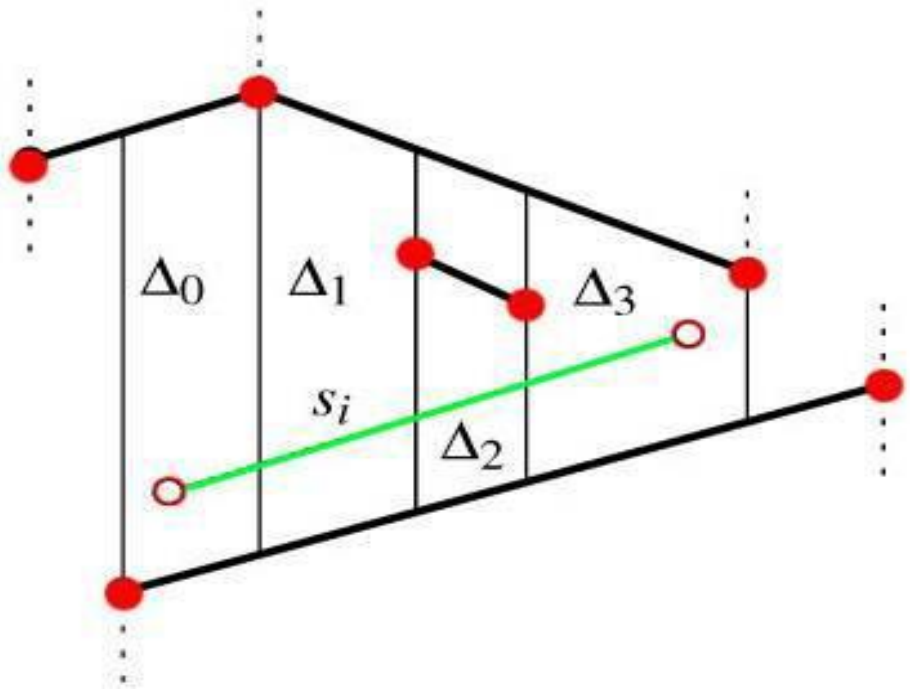
Algorithm TRAPEZOIDALMAP(S)

Input. A set S of n non-crossing line segments.

Output. The trapezoidal map $\mathcal{T}(S)$ and a search structure $\mathcal{D}$ for $\mathcal{T}(S)$ in a bounding box.

1. Determine a bounding box R that contains all segments of S, and initialize the trapezoidal map structure $\mathcal{T}$ and search structure $\mathcal{D}$ for it.
2. Compute a random permutation $s_1, s_2, \ldots, s_n$ of the elements of S.
3. **for** $i \leftarrow 1$ **to** n
4. **do** Find the set $\Delta_0, \Delta_1, \ldots, \Delta_k$ of trapezoids in $\mathcal{T}$ properly intersected by s_i. → **Algorithm** FOLLOWSEGMENT($\mathcal{T}, \mathcal{D}, s_i$)
5. Remove $\Delta_0, \Delta_1, \ldots, \Delta_k$ from $\mathcal{T}$ and replace them by the new trapezoids that appear because of the insertion of s_i.
6. Remove the leaves for $\Delta_0, \Delta_1, \ldots, \Delta_k$ from $\mathcal{D}$, and create leaves for the new trapezoids. Link the new leaves to the existing inner nodes by adding some new inner nodes, as explained below.

بررسی الگوریتم

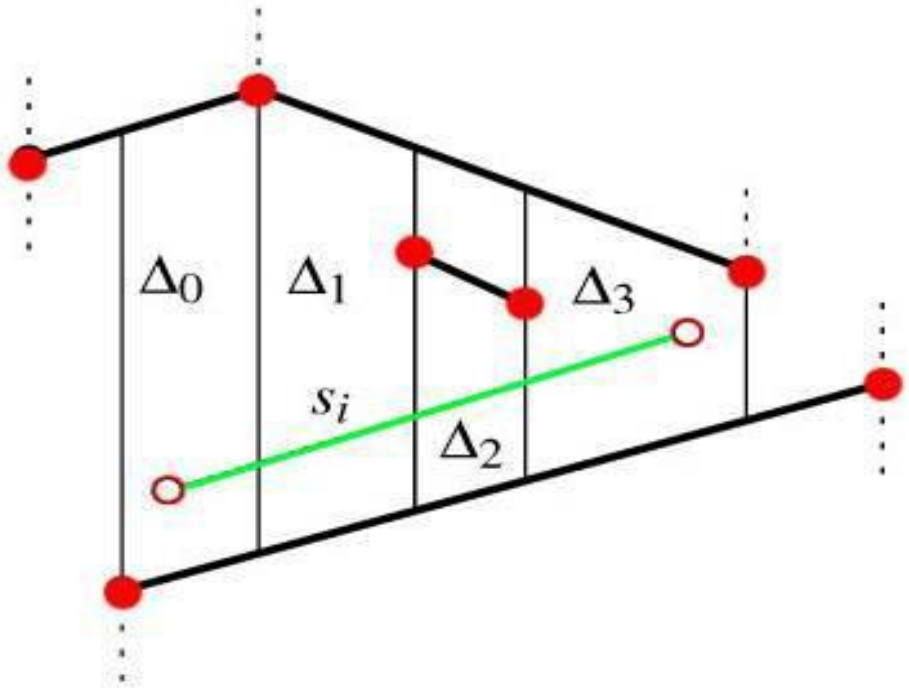


هرذوزنقه Δ_{j+1} همسایه سمت راست Δ_j هست:

اگر rightp(Δ_j) بالای s_i قرار بگیرد Δ_{j+1} همسایه پایین راست Δ_j می باشد

اگر rightp(Δ_j) پایین s_i قرار بگیرد Δ_{j+1} همسایه بالا راست Δ_j می باشد

بررسی الگوریتم



فرض کنید $S_i=\{s_1 , s_2 , \dots ,s_i\}$ که s_i ها به صورت تصادفی درج می شوند.

در ابتدا $T=T(S_0)=T(\emptyset)$ شامل تنها یک ذوزنقه هست که همان مستطیل R می باشد.

بعد از اضافه کردن پاره خط s_i افراز $T(S_{i-1})$ به $T(S_i)$ تبدیل می گردد.

$\Delta_0,\Delta_1,\dots,\Delta_k$ ذوزنقه هایی از $T(S_{i-1})$هستند که s_i آنها را قطع می کند.

با توجه به این همسایگی فقط کافی است ذوزنقه Δ_0 که راس انتهایی چپ s_i (p) در آن قرار دارد مشخص شود و بقیه ذوزنقه ها با بررسی همسایگی Δ_0 بدست می آید.

نحوه پیدا کردن Δ_0، از طریق جستجو در ساختار D:

اگر p در ساختار D حضور نداشته باشد که با جستجو به ذوزنقه حاوی p خواهیم رسید.

اگر p در ساختار D حضور داشته باشد دو حالت وجود دارد:

یا p روی پاره خط عمودی در یک x-راس باشد، p را سمت راست در نظر می گیریم.

یا p روی یک پاره خط s از یک y-راس باشد، شیب s و s_i را مقایسه میکنیم:

اگر شیب s_i بزرگتر بود پس p بالای s قرار دارد و بالعکس.

پایان