

Point Visibility Region

Anteriormente discutimos o problema de determinação da visibilidade de um ponto x encerrado no interior de um polígono.

Vamos, agora, mudar este ambiente e, em lugar de termos uma sequência de segmentos ordenadamente conectados em ~~entre~~ seus extremos, cercando o ponto x , teremos um conjunto de segmentos que se interceptam, no máximo, em seus extremos, reduzindo o campo de visão do ponto x . Não teremos mais, portanto, uma região de visibilidade fechada, mas, possivelmente, uma região potencialmente ilimitada.

Definição do Problema

Seja um conjunto de segmentos S que se interceptam, no máximo, em seus extremos e um ponto x . Queremos determinar $V(x)$, a região de visibilidade do ponto x .

Cota Inferior

Antes de apresentarmos um algoritmo para a resolução do problema, vamos estabelecer uma cota inferior [0'Reurte], a fim de que dirijamos nossos esforços para alcançá-la.

A seguir determinaremos a cota inferior $\mathcal{R}(n \log n)$ para redução do problema de ordenação de m números inteiros.

Leia uma sequência $S = (x_1, \dots, x_m)$ de valores inteiros. Vouremos a partir da sequência S , construir um polígono com braços (como esquema do problema) que represente uma instância do problema de visibilidade.

Defina x_{\max} e x_{\min} como sendo o maior e menor valor, respectivamente, de S , também defina $\Delta = x_{\max} - x_{\min}$.

O polígono é um retângulo cujos vértices estão localizados nos pontos $(x_{\min}-1, -\Delta/2)$, $(x_{\max}+1, -\Delta/2)$, $(x_{\max}+1, \Delta/2)$ e $(x_{\min}-1, \Delta/2)$.

A cada ponto x_i ($1 \leq i \leq m$) é associado um braço retangular com vértices em $(x_i - \varepsilon, -\varepsilon)$, $(x_i + \varepsilon, -\varepsilon)$, $(x_i + \varepsilon, \varepsilon)$ e $(x_i - \varepsilon, \varepsilon)$, $\varepsilon = 0.1$, por exemplo.

O ponto x , a partir do qual a visibilidade é tomada, é tomado no canto inferior esquerdo do triângulo, ou seja, $x = (x_{\min}-1, -\Delta/2)$. É facilmente percebido que o vértice inferior esquerdo de cada braço ocorre a cada cinco vértices, conforme x cresce. Assim, é fácil extrair a sequência ordenada de x_i 's a partir de um algoritmo que produza a fronteira de $V(x)$ como uma sequência de vértices.

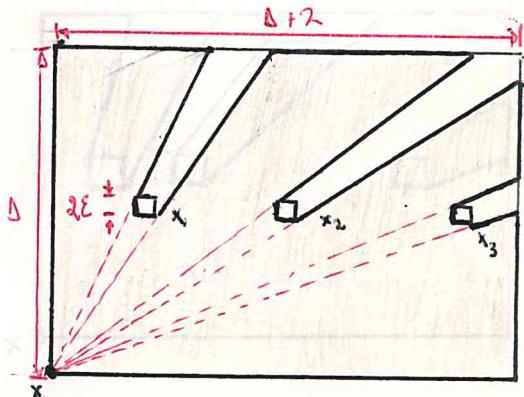


fig 1

O Algoritmo: descrição informal

O algoritmo consiste de uma varredura planar de um rato r em forma do ponto x (fig. 2).

Seja o conjunto de segmentos $S = \{s_1, \dots, s_n\}$ e um ponto x , tal que s_k ($1 \leq k \leq n$) é determinado pelos vértices v_{kini} e v_{kfin} .

Vamos ordenar os vértices v_i ($1 \leq i \leq 2k$), segundo seu ângulo em torno do ponto x , a partir de um vértice especial que apresentaremos mais tarde (fig 3).

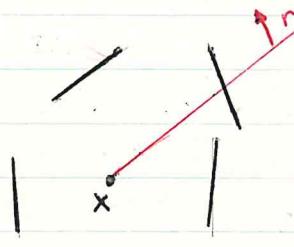


fig 2

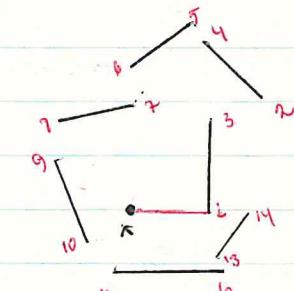


fig 3

Percorreremos estes vértices na ordem estabelecida atualizando as seguintes estruturas:

- conjunto² de segmentos ativos: armazena o conjunto de segmentos de S cujos vértices iniciais foram visitados, mas não tiveram seus vértices finais encontrados. Entre São os segmentos que correntemente podem obter a vizinhança de x ;

- biombo: determina um sub-segmento de algum dos segmentos ativos, que efetua-

1 - v_{kini} e v_{kfin} são tais que $x v_{kini} v_{kfin}$ é "virada à esquerda"

2 - neste momento não obviamente particularizar a estrutura v_{kfin} .

4

mente estático obscurecendo a visão do ponto x num dado instante;

• conjunto de segmentos que determinam a região de visibilidade, isto é, a resposta do processamento.

Quando encontramos um vértice de vértice de um de segmento temos duas situações possíveis:

1º o vértice v está "atrás" de algum dos segmentos ativos (fig 4). Neste caso, apenas inserimos o segmento D_v , associado ao vértice v , no conjunto de segmentos ativos, pois, provavelmente, o segmento (ou os segmentos) que o obstruiria tem seus extremos finais encontrados e D_v torna-se "o mais à frente" entre os segmentos ativos;



fig 4

2º o vértice v está "na frente" dos segmentos ativos (fig 5). Neste caso, o novo segmento D_v que se inicia, esconde parte do plano σ , se extinguir algum segmento, esconderá parte do segmento ativo mais próximo de x (biombo) que, antes de examinarmos v , era o segmento ativo a partir de x que limitava porção do plano. Sejam: $\bar{\sigma}$, o segmento do biombo; b_{ini} , o vértice do início do biombo; y , a intersecção do reto de terminar por xv com o segmento $\bar{\sigma}$. Vamos inserir os segmentos $[b_{ini}, y]$ e $[y, v]$ no conjunto de segmentos resposta, para eles dividirem o

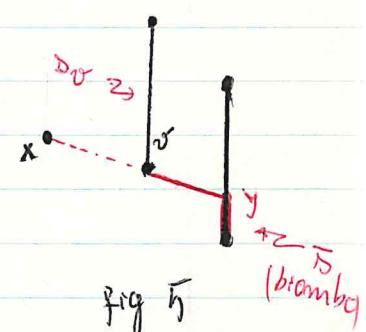


fig 5

plano em uma região visível e outra invisível a partir de x .

Como o segmento s_v encerrou parte dos segmentos outros, ele determinará um novo brombo com vértice em v_i .

Por outro lado, ao encontrarmos um vértice v_i de fim de segmento, temos, também, duas possibilidades:

- o vértice v_i está "atrás" de algum dos segmentos outros, sendo, portanto, invisível a partir do ponto x . O segmento s_v , encerrado por v_i , não influirá mais na visibilidade do ponto x (fig 6). Ele é, simplesmente, desativado;

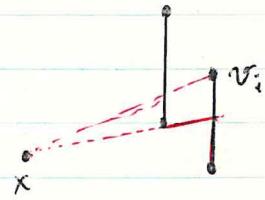


fig 6

- o vértice v_i está na "frente" dos segmentos outros (fig 7), ou seja, é vértice final do ~~brombo~~ segmento do brombo corrente. O segmento s_v , cujo extremo final é v_i , é o segmento que momentaneamente está obstruindo a visão do ponto x . Portanto, então, além de desativar o segmento, inibir o segmento [brombo, ini, vi] e o segmento $[v_i, y]$, tal que y é a intersecção do rayo determinado por x, v_i e o segmento s_y . "imediatamente" atrás de s_v (caso exibido)

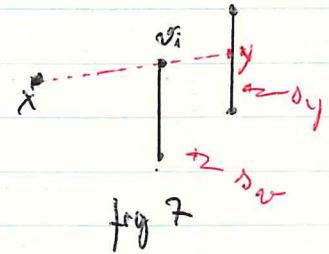


fig 7

¹ Veremos que o vértice do brombo não é necessariamente algum vértice de segmento. Pode ser qualquer ponto do segmento do brombo.

O segmento, só determinaria o novo
braço com ponto inicial em y .

Note que enunciou que o vértice de
fim de segmento, quemelo nível percorre
exatamente, segmento que determina o braço
corrente. Isto é garantido pelo fato que os
segmentos se interceptam, no máximo, em
seus extremos.

O Algoritmo e as Estruturas de Dados: descrição detalhada

Na seção anterior fizemos uma descri-
ção superficial do algoritmo. Intensificamos,
agora, começar detalhes do algoritmo e das
estruturas de dados que utilizaremos para ob-
tengor a cota inferior do algoritmo.

Lembre-se: vamos ordenar os vértices
segundo seu ângulo em torno do ponto x ,
e, conforme os percorremos nessa ordem, dire-
mos, de forma eficiente, poder ativar ou
desativar segmentos aos quais pertencem e
poder determinar o segmento entre mais
próximo do ponto x .

Para não extraparmos a cota inferior
deveremos garantir que, para cada ponto, estes
operações não excedam tempo $O(\log n)$.

Vamos, portanto, utilizar uma árvore AVL
de segmentos abertos (SegsAVL) organizada
de forma que dados dois segmentos, s_t
e s_r , tal que o não correspondente a s_t é
anterior ao não correspondente a s_r , e um
raio r (com origem em x) que intercepta

$\mathcal{D}_f \in \mathcal{O}_r$ em Z_f e Z_r , respectivamente, \mathcal{D}_r entaréi men sub-árvore direita de \mathcal{D}_f se $|x_{\mathbb{Z}_r}| > |x_{Z_f}|$.

A árvore, assim estruturada, permitirá que em segmento seja atendido o desafio do tempo $O(\log n)$, bem como obter uma ponte p , vizinha a partir de x , obtemos "imediatamente" o segmento envolto "imediatamente" à esquerda dele, em tempo $O(\log n)$ (i.e. o segmento mais à esquerda da sub-árvore esquerda de $\mathcal{D}_{\text{left}}$).

A seguir, vou resumidamente apresentar os principais estruturas de dados utilizadas no algoritmo:

1. `Vertice` (`tipo`, `pSeg`, `ponte`, `codigo`)

Um vértice contém as seguintes informações:

- `tipo`: inicial ou final;
- `pSeg`: apontador p/ o segmento a que pertence;
- `ponte`: a coordenada (x, y) do vértice;
- `codigo`: main é frenzí numero que vértices podem ser errados! Este código informa se este vértice é um vértice original, ou é um vértice em troca e faz parte de um retângulo que mantemos no conjunto de segmentos.

A ordenação dos vértices de feita pelas
seguintes regras (em ordem decrescente de prioridade):

- 1) $v_i < v_j$, se $d(v_i) < d(v_j)$;
- 2) $v_i < v_j$, se $d(x, v_i) > d(x, v_j)$
- 3) $v_i < v_j$, se v_i é vértice de inicio do segmento e v_j é vértice de final do segmento;
- 4) caso contrário $v_i = v_j$.

2. segmento (ponto-inicial, ponto-final)

Um segmento é determinado por um ponto inicial e um ponto-final, ou forma uma figura "ponto-inicial ponto-final" que "vira à esquerda".

3. bimbo (bifurcação).

O bimbo contém o ponto inicial do bimbo determinado por algum segmento.

4. Seções.

A árvore de segmentos ou árvore contém em cada nó um segmento e um apontador para suas sub-árvore esquerda e/ou sub-árvore

- 1- o segundo critério não é aplicado se ocorrer empate no 1º critério.
- 2- $\alpha(v)$: ângulo em termo do ponto x.

esquerda.

Sóhne éta avrone náe permitidas as
seguintes operações:

- Ativa Segmento (segmento)
- Desativa Segmento (segmento)
- Seg Ativo Mais proximo(s) retorna o segmento ativo imediatamente atrás da vértice v .

A fim de simplificar o algoritmo
vamos fazer algumas modificações nas instâncias
do problema.

- no conjunto S de vértices adjacentes
9 segmentos que determinam um retângulo
que envolve todos os segmentos de S . Isso
pode ser feito em tempo linear e garantir
que o raro r sempre intercepta algum
segmento. Vértices que, por sua vez,
pertencem a esses segmentos serão denominados vértices
do retângulo;
- temos errar um vértice inicial v_0
adequado, a partir do qual vértices ordenados
os outros vértices (em regra, v_0 tem ângulos 0).

A partir de x traçar um raro r . Os
segmentos s que r interceptar em pontos y
serão divididos em dois segmentos s_i e s_f ,
tal que s_i estará à esquerda do raro
 r (orientado de x para v_0). Os novos segmentos

SOO mineralizados da seguinte forma (fig 8):

$$\begin{aligned} D_{i,i} \text{mi} &= y; & D_{i,i} \text{firm} &= S_i \text{ firm} \\ D_{f,i} \text{mi} &= D_{-i} \text{mi}; & D_{f,-i} \text{firm} &= y \end{aligned}$$

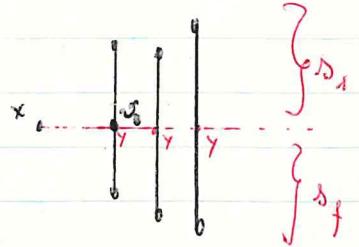


Fig 8

O vértice v_0 será o vértice mineral mais próximo de x , pertencente a algum dos segmentos S_i .

Os segmentos só devem ser armazenados, para serem tratados imediatamente.

Enta adaptação é necessária pois, como podemos notar na figura 9, qualquer um dos vértices que tomarmos como vértice inicial, ou obscuridade, ou é obscurecido pelo segmento anterior, do qual ainda não se tem consciência, quando tal segmento for analisado, o segmento que contém o vértice mineral já foi tratado, não tendo-se, portanto, consciência dele. Estes dois fatos levaram as exceções equivocadas do algoritmo.

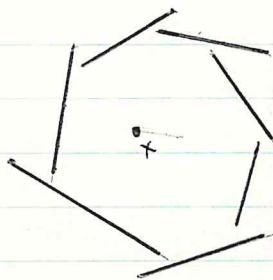


Fig 9

Algoritmo

Crie o vértice v_0 conforme o segmento descrito
Lembrando dos segmentos errados /

Inicia o retângulo

Extrair e ordenar os vértices segundo o
entorno estabelecido

Ative os segmentos lembrados

bombo $\leftarrow v_0$

para todo vértice v_i

se v_i é vértice de ímico
então se v_i não é antífronal /* caso 1a */
então Ativa Segmento (v_i , pleg)
fim se

se v_i não está encoberto
então $r \leftarrow \text{ray}(x, v_i)$ /* caso 1b */
 $s \leftarrow \text{Seg Ativo Mais Próximo}()$
 $y \leftarrow \text{intersecção}(r, s)$
 $\text{inser}([bombo, y])$
 $\text{inser}([y, v_i])$
 $bombo \leftarrow v_i$
fim se

senão /* Vértice de fim */

se vértice está encoberto
então Remove Segmento (v_i , pleg) /* caso 2a */
senão /* Ref 1 */
fim se

fim se

fim para
fim algoritmo

Ref 1 /* v_i é final e v_{i+1} */

Derativa Segmento (v_i, pSeg)

r = raro (x, v_i)

y = intersecção (r, s)

insera ([briombo, v_i])

~~insera ([briombo,~~

insera ([v_i, y])

briombo \leftarrow y

/* vértice final
do retângulo resultante
está encoberto por
vertice inicial */

fim ref

Obs-:

* insera ([a, b]) significa inverte o segmento
com extremo inicial em a e
final em b no conjunto de
segmentos de resposta.