Geometria Computacional

Cristina G. Fernandes

Departamento de Ciência da Computação do IME-USP

`http://www.ime.usp.br/~cris/`

segundo semestre de 2009

Algoritmo de Shamos e Hoey

Entrada: coleção $e[1 \mathinner{.\,.} n], d[1 \mathinner{.\,.} n]$ de segmentos.

Saída: VERDADE se há dois segmentos na coleção que se intersectam, e FALSO caso contrário.

Algoritmo de Shamos e Hoey

Entrada: coleção $e[1\,..\,n], d[1\,..\,n]$ de segmentos.

Saída: VERDADE se há dois segmentos na coleção que se intersectam, e FALSO caso contrário.

Hipótese simplificadora:

Não há dois pontos extremos com a mesma X-coordenada.

Em particular, não há segmentos verticais,
nem dois segmentos com extremos coincidentes.

Algoritmo de Shamos e Hoey

INTERSEÇÃO-SH(e, d, n)

$(E, segm, esq) \leftarrow$ FILADEEVENTOS(e, d, n)
CRIE(T)
para $p \leftarrow 1$ até $2n$ faça
    $i \leftarrow segm[p]$
    $pred \leftarrow$ PREDECESSOR$(T, E_X[p], E_Y[p])$
    $suc \leftarrow$ SUCESSOR$(T, E_X[p], E_Y[p])$
    se $esq[p]$
        então INSIRA(T, i)
            se $(pred \neq$ NIL e INTER$(e, d, i, pred))$
            ou $(suc \neq$ NIL e INTER$(e, d, i, suc))$
                então devolva VERDADE
        senão REMOVA(T, i)
            se $pred \neq$ NIL e $suc \neq$ NIL e INTER$(e, d, pred, suc)$
                então devolva VERDADE
devolva FALSO

Cuidados especiais

Hipótese simplificadora:

Não há dois pontos extremos com a mesma X-coordenada.

Em particular, não há segmentos verticais,
nem dois segmentos com extremos coincidentes.

Como nos livramos desta hipótese?

Cuidados especiais

Hipótese simplificadora:

Não há dois pontos extremos com a mesma X-coordenada.

Em particular, não há segmentos verticais,
nem dois segmentos com extremos coincidentes.

Como nos livramos desta hipótese?

Pontos extremos com mesma X-coordenada:

Se existir um segmento vertical,
deixe o extremo superior no vetor e e o inferior no vetor d.

Cuidados especiais

Hipótese simplificadora:

Não há dois pontos extremos com a mesma X-coordenada.

Em particular, não há segmentos verticais,
nem dois segmentos com extremos coincidentes.

Como nos livramos desta hipótese?

Pontos extremos com mesma X-coordenada:

Se existir um segmento vertical,
deixe o extremo superior no vetor e e o inferior no vetor d.

Se os segmentos podem ter extremos em comum,
então os extremos em e devem aparecer em E
antes dos extremos em d.

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Você consegue projetar um algoritmo que consuma tempo $\mathrm{O}(n \lg n)$ para este problema?

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Você consegue projetar um algoritmo que consuma tempo $\mathrm{O}(n \lg n)$ para este problema?

No máximo, quantos pares teremos que imprimir?

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Você consegue projetar um algoritmo
que consuma tempo $\mathrm{O}(n \lg n)$ para este problema?

No máximo, quantos pares teremos que imprimir?

Algoritmos sensíveis à saída (*output sensitive*).

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Como adaptar o algoritmo de Shamos e Hoey?

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Como adaptar o algoritmo de Shamos e Hoey?

Novo tipo de ponto evento: as interseções.

Como tratá-las?

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Como adaptar o algoritmo de Shamos e Hoey?

Novo tipo de ponto evento: as interseções.

Como tratá-las?

Ao detectar cada uma, além de imprimi-la,
a colocamos na fila de eventos (que é agora dinâmica).

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Como adaptar o algoritmo de Shamos e Hoey?

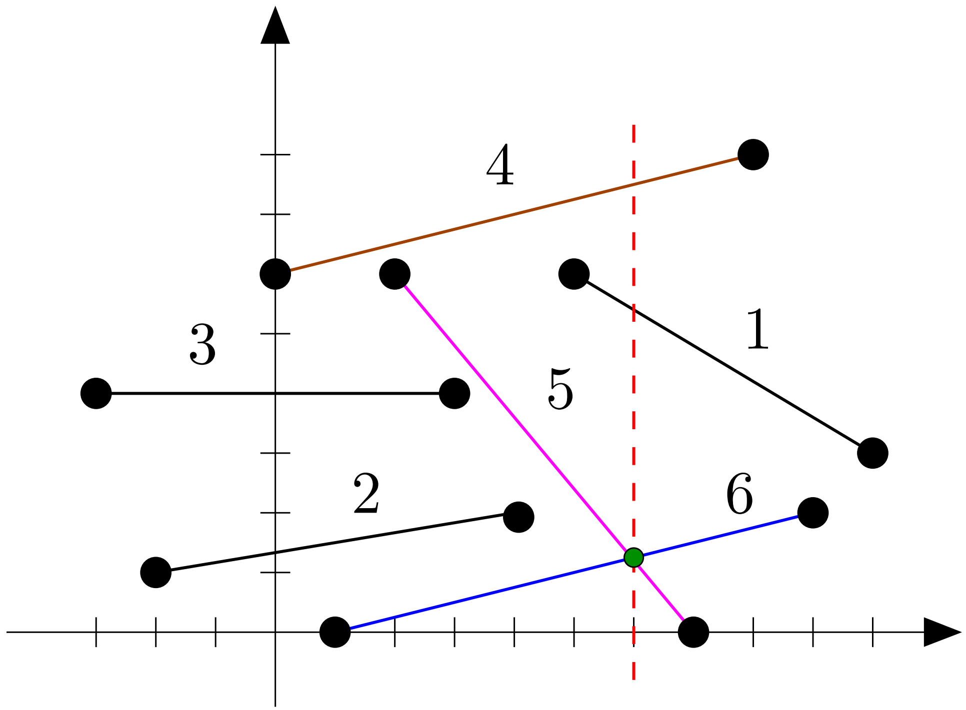
Novo tipo de ponto evento: as interseções.

Como tratá-las?

Ao detectar cada uma, além de imprimi-la,
a colocamos na fila de eventos (que é agora dinâmica).

Ao processar um ponto evento que é uma interseção, deve-se inverter a ordem dos segmentos que se intersectam neste ponto.

Ponto evento: interseção



Antes do ponto evento: $4 \prec 1 \prec 5 \prec 6$

Depois do ponto evento: $4 \prec 1 \prec 6 \prec 5$

Algoritmo de Bentley e Ottmann

Entrada: coleção $e[1 \,.\,.\, n], d[1 \,.\,.\, n]$ de segmentos.

Algoritmo de Bentley e Ottmann

Entrada: coleção $e[1\,..\,n], d[1\,..\,n]$ de segmentos.

Saída: todos os pares de segmentos da coleção que se intersectam.

Algoritmo de Bentley e Ottmann

Entrada: coleção $e[1\,..\,n], d[1\,..\,n]$ de segmentos.

Saída: todos os pares de segmentos da coleção que se intersectam.

Hipótese simplificadora:

Não há dois pontos eventos com a mesma X-coordenada.

Em particular, não há interseção com mesma X-coordenada que outra, ou que algum extremo de segmento.

Algoritmo de Bentley e Ottmann

Entrada: coleção $e[1\,.\,.\,n], d[1\,.\,.\,n]$ de segmentos.

Saída: todos os pares de segmentos da coleção que se intersectam.

Hipótese simplificadora:

Não há dois pontos eventos com a mesma X-coordenada.

Em particular, não há interseção com mesma X-coordenada que outra, ou que algum extremo de segmento.

Não há interseções múltiplas, ou seja, não há um ponto em mais do que dois segmentos da coleção.

Fila de eventos

Agora ela é dinâmica: sofre inserções (e, como antes, remoções).

Que ED usar para a fila de eventos?

Fila de eventos

Agora ela é dinâmica: sofre inserções
(e, como antes, remoções).

Que ED usar para a fila de eventos?

ABBB com ordem dada pelas X-coordenadas dos pontos.

Fila de eventos

Agora ela é dinâmica: sofre inserções
(e, como antes, remoções).

Que ED usar para a fila de eventos?

ABBB com ordem dada pelas X-coordenadas dos pontos.

A fila começa com os extremos dos intervalos.

A cada iteração,
removemos um evento da fila para processá-lo.

Fila de eventos

Agora ela é dinâmica: sofre inserções
(e, como antes, remoções).

Que ED usar para a fila de eventos?

ABBB com ordem dada pelas X-coordenadas dos pontos.

A fila começa com os extremos dos intervalos.

A cada iteração,
removemos um evento da fila para processá-lo.

Ao detertar uma interseção,
inserimos tal ponto na fila de eventos.

Quantos elementos estão na fila no pior caso?

Montagem da fila de eventos

FILADEEVENTOS:
recebe $e[1 \,..\, n]$ e $d[1 \,..\, n]$ com extremos dos segmentos

troca $e[i]$ por $d[i]$ para todo i tal que $e_X[i] < d_X[i]$
($e[i]$: extremo esquerdo do segmento i e $d[i]$ o direito)

Montagem da fila de eventos

FILADEEVENTOS:
recebe $e[1\,..\,n]$ e $d[1\,..\,n]$ com extremos dos segmentos

troca $e[i]$ por $d[i]$ para todo i tal que $e_X[i] < d_X[i]$
($e[i]$: extremo esquerdo do segmento i e $d[i]$ o direito)

devolve apontador F para ABBB contendo os pontos de $e[1\,..\,n]$ e $d[1\,..\,n]$ ordenados por X-coordenada.

Montagem da fila de eventos

FILADEEVENTOS:
recebe $e[1 \mathinner{.\,.} n]$ e $d[1 \mathinner{.\,.} n]$ com extremos dos segmentos

troca $e[i]$ por $d[i]$ para todo i tal que $e_X[i] < d_X[i]$
($e[i]$: extremo esquerdo do segmento i e $d[i]$ o direito)

devolve apontador F para ABBB contendo os pontos de $e[1 \mathinner{.\,.} n]$ e $d[1 \mathinner{.\,.} n]$ ordenados por X-coordenada.

Cada nó q desta ABBB contém campos
$evento(q)$: o ponto evento
$tipo(q)$: esq/dir se é extremo esquerdo/direito de segmento
$inter$ se é interseção de dois dos segmentos

Montagem da fila de eventos

FILADEEVENTOS:
recebe $e[1\,..\,n]$ e $d[1\,..\,n]$ com extremos dos segmentos

troca $e[i]$ por $d[i]$ para todo i tal que $e_X[i] < d_X[i]$
($e[i]$: extremo esquerdo do segmento i e $d[i]$ o direito)

devolve apontador F para ABBB contendo os pontos de $e[1\,..\,n]$ e $d[1\,..\,n]$ ordenados por X-coordenada.

Cada nó q desta ABBB contém campos
$evento(q)$: o ponto evento
$tipo(q)$: esq/dir se é extremo esquerdo/direito de segmento
$inter$ se é interseção de dois dos segmentos

$segm(q)$: índice do segmento que tem esse ponto como extremo, ou, se é uma interseção, índices dos dois segmentos que o contém.

Algoritmo de Bentley e Ottmann

INTERSEÇÃO-BO(e, d, n)

$(F, \mathit{evento}, \mathit{tipo}, \mathit{segm}) \leftarrow$ FILADEEVENTOS(e, d, n)
CRIE(T)
enquanto não VAZIA(F) faça
    $q \leftarrow$ REMOVAMIN(F)
    se $\mathit{tipo}(q) \neq \mathit{inter}$
        então $i \leftarrow \mathit{segm}(q)$
            $\mathit{pred} \leftarrow$ PREDECESSOR$(T, e_X[i], e_Y[i])$
            $\mathit{suc} \leftarrow$ SUCESSOR$(T, e_X[i], e_Y[i])$
            se $\mathit{tipo}(q) = \mathit{dir}$
                então REMOVA(T, i)
                    se $\mathit{pred} \neq$ NIL e $\mathit{suc} \neq$ NIL e INTER$(e, d, \mathit{pred}, \mathit{suc})$
                        então $(x, y) \leftarrow$ CALCULEINTER$(e, d, \mathit{pred}, \mathit{suc})$
                            INSIRA$(F, x, y, \mathit{pred}, \mathit{suc})$
                senão INSIRA(T, i)
                    $\ldots$

Algoritmo de Bentley e Ottmann

INTERSEÇÃO-BO(e, d, n)

$(F, \mathit{evento}, \mathit{tipo}, \mathit{segm}) \leftarrow$ FILADEEVENTOS(e, d, n)
CRIE(T)
enquanto não VAZIA(F) faça
    $q \leftarrow$ REMOVAMIN(F)
    se $\mathit{tipo}(q) \neq \mathit{inter}$
        então $\cdots$
        senão $(i, j) \leftarrow \mathit{segm}(q)$
            TROQUE(T, i, j)
            $(x, y) \leftarrow \mathit{evento}(q)$
            $\mathit{pred} \leftarrow$ PREDECESSOR(T, x, y)
            $\mathit{suc} \leftarrow$ SUCESSOR(T, x, y)
            se $\mathit{pred} \neq$ NIL e INTER(e, d, pred, j)
                então $\cdots$
            se $\mathit{suc} \neq$ NIL e INTER(e, d, suc, i)
                então $\cdots$

Consumo de tempo

Seja I o número de interseções.

O algoritmo executa $2n + I$ iterações.

Consumo de tempo

Seja I o número de interseções.
O algoritmo executa $2n + I$ iterações.

Cada iteração faz uma chamada a PREDECESSOR, SUCESSOR, e uma a INSIRA ou REMOVA, na ABBB T.

Na ABBB T, em qualquer momento, há $\mathrm{O}(n)$ segmentos.

Assim, cada operação destas consome tempo $\mathrm{O}(\lg n)$.

Consumo de tempo

Seja I o número de interseções.
O algoritmo executa $2n + I$ iterações.

Cada iteração faz uma chamada a PREDECESSOR, SUCESSOR, e uma a INSIRA ou REMOVA, na ABBB T.

Na ABBB T, em qualquer momento, há $\mathrm{O}(n)$ segmentos.

Assim, cada operação destas consome tempo $\mathrm{O}(\lg n)$.

Cada iteração faz uma chamada a REMOVAMIN e, eventualmente, uma a INSIRA na ABBB F.

Na ABBB F, em qq momento, há $\mathrm{O}(n + I) = \mathrm{O}(n^2)$ pontos.

Assim, cada operação consome tempo $\mathrm{O}(\lg n^2) = \mathrm{O}(\lg n)$.

Consumo de tempo

Seja I o número de interseções.
O algoritmo executa $2n + I$ iterações.

Cada iteração faz uma chamada a PREDECESSOR, SUCESSOR, e uma a INSIRA ou REMOVA, na ABBB T.

Na ABBB T, em qualquer momento, há $\mathrm{O}(n)$ segmentos.

Assim, cada operação destas consome tempo $\mathrm{O}(\lg n)$.

Cada iteração faz uma chamada a REMOVAMIN e, eventualmente, uma a INSIRA na ABBB F.

Na ABBB F, em qq momento, há $\mathrm{O}(n + I) = \mathrm{O}(n^2)$ pontos.

Assim, cada operação consome tempo $\mathrm{O}(\lg n^2) = \mathrm{O}(\lg n)$.

As demais operações efetuadas em uma iteração consomem tempo $\mathrm{O}(1)$ (mesmo as chamadas a INTER).

Consumo de tempo

Seja I o número de interseções.
O algoritmo executa $2n + I$ iterações.

Cada iteração faz uma chamada a PREDECESSOR, SUCESSOR, e uma a INSIRA ou REMOVA, na ABBB T.

Na ABBB T, em qualquer momento, há $\mathrm{O}(n)$ segmentos.

Assim, cada operação destas consome tempo $\mathrm{O}(\lg n)$.

Cada iteração faz uma chamada a REMOVAMIN e, eventualmente, uma a INSIRA na ABBB F.

Na ABBB F, em qq momento, há $\mathrm{O}(n + I) = \mathrm{O}(n^2)$ pontos.

Assim, cada operação consome tempo $\mathrm{O}(\lg n^2) = \mathrm{O}(\lg n)$.

O consumo de tempo por iteração é $\mathrm{O}(\lg n)$, e o algoritmo de Bentley e Ottmann consome tempo $\mathrm{O}((n + I) \lg n)$.

Cuidados especiais

Hipótese simplificadora:

Não há dois pontos eventos com a mesma X-coordenada.

Em particular, não há interseção com mesma X-coordenada que outra, ou que algum extremo de segmento.

Não há interseções múltiplas, ou seja, não há um ponto em mais do que dois segmentos da coleção.

Como nos livramos destas hipóteses?

Cuidados especiais

Hipótese simplificadora:

Não há dois pontos eventos com a mesma X-coordenada.

Em particular, não há interseção com mesma X-coordenada que outra, ou que algum extremo de segmento.

Não há interseções múltiplas, ou seja, não há um ponto em mais do que dois segmentos da coleção.

Como nos livramos destas hipóteses?

O mais complicado é tratar das interseções múltiplas.
Isso está detalhado no livro de de Berg e outros, capítulo 2.

Cuidados especiais

Hipótese simplificadora:

Não há dois pontos eventos com a mesma X-coordenada.

Em particular, não há interseção com mesma X-coordenada que outra, ou que algum extremo de segmento.

Não há interseções múltiplas, ou seja, não há um ponto em mais do que dois segmentos da coleção.

Como nos livramos destas hipóteses?

O mais complicado é tratar das interseções múltiplas.
Isso está detalhado no livro de de Berg e outros, capítulo 2.

Agora... vamos ver uma animação!