Decomposição Trapezoidal

Claudio Esperança
Paulo Roma

Localização no Plano

- Vimos a estrutura de Kirkpatrick que permite localização de pontos numa triangulação
  - Consulta em tempo $O(\log n)$ ☺
  - Construção em $O(n \log n)$ ☺
  - Espaço $O(n)$ ☺
  - Não muito prática ☹
    - Constantes altas
- Mapas trapezoidais são uma alternativa interessante para resolver o problema

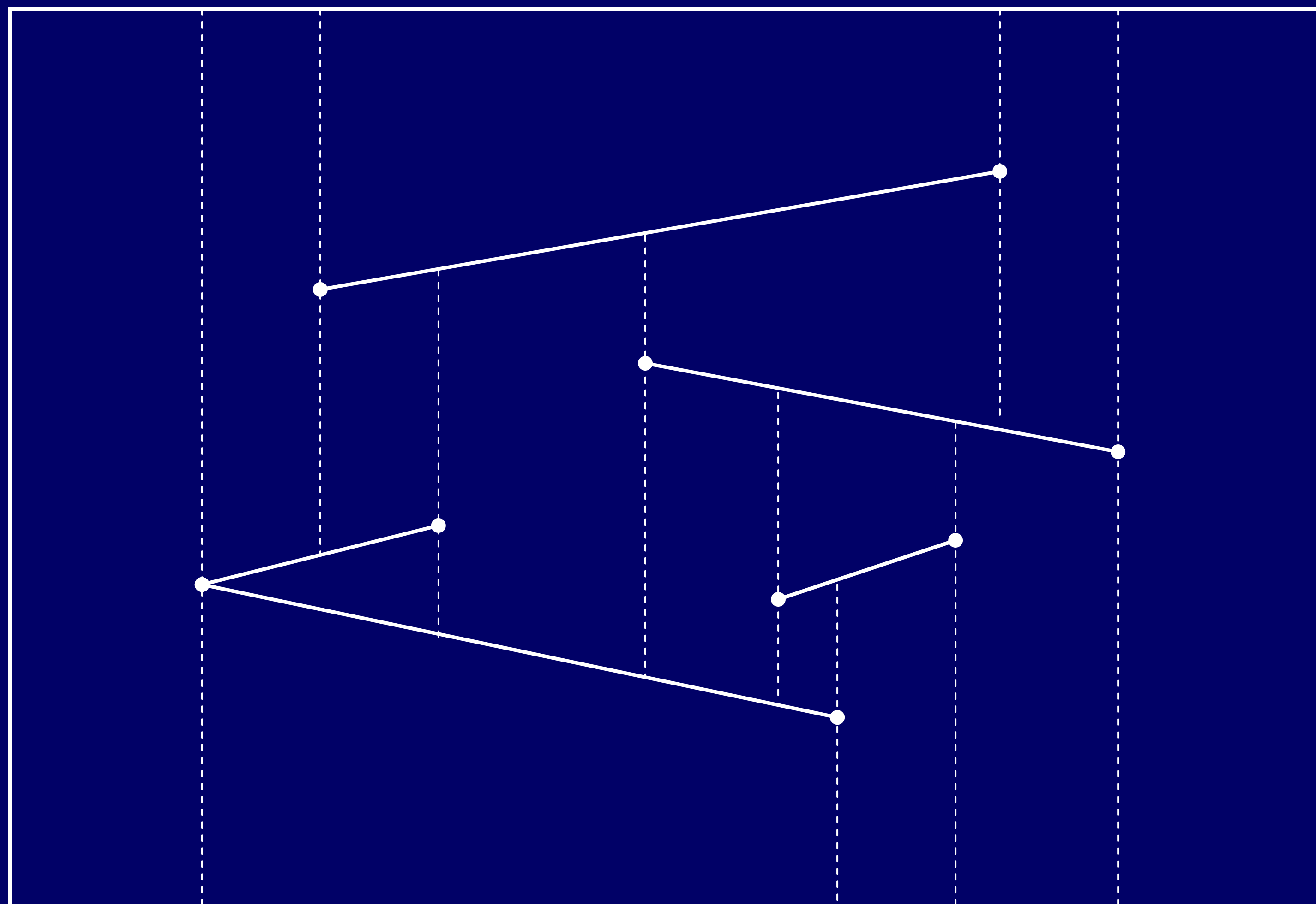
Mapas Trapezoidais

- Mesmas complexidades de caso médio
- Facilmente adaptada para tratar com subdivisões poligonais quaisquer
  - Arestas dos polígonos são inseridas uma a uma
- Construção randomizada
  - Independe da geometria ou da consulta
  - Depende apenas da ordem de inserção das arestas
- Mais prática que a estrutura de Kirkpatrick

Mapas Trapezoidais

- É uma divisão do plano induzida por uma coleção de segmentos de reta
  - Assume-se que não há segmentos verticais
  - Segmentos não se interceptam a não ser em suas extremidades
  - Lados esquerdo e direito de cada trapézio apoiam-se em extremidades dos segmentos
  - Cada extremidade "atira uma bala" para cima e outra para baixo até encontrar outro segmento de reta
  - Para evitar arestas infinitas, cria-se um retângulo contendo todas os segmentos

Mapas Trapezoidais



Conversão de mapas poligonais

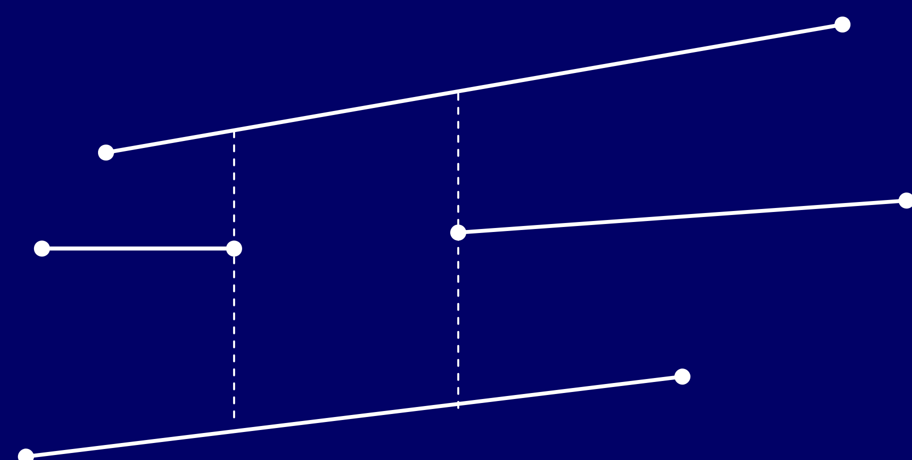
- Cada extremidade dispara 2 balas que irão atingir outros segmentos criando 2 novos vértices
  - Cada vértice do mapa poligonal gera 3 vértice no mapa trapezoidal
  - Se temos n segmentos de reta, o mapa trapezoidal terá no máximo <u>$6n + 4$ vértices</u> (contando os 4 vértices do retângulo envolvente)

Conversão de mapas poligonais

- Cada trapezóide é limitado à esquerda e à direita por alguma extremidade de algum segmento de reta original
  - A extremidade esquerda de cada segmento limita a aresta esquerda de 2 trapézios (acima e abaixo)
  - A extremidade direita de cada segmento limita a aresta esquerda de 1 trapézio (à direita do segmento)
  - Portanto, cada segmento corresponde a 3 trapézios e teremos <u>$3n + 1$ trapézios</u> no total
    - O trapézio adicional é corresponde ao lado esquerdo do retângulo envolvente

Algumas observações

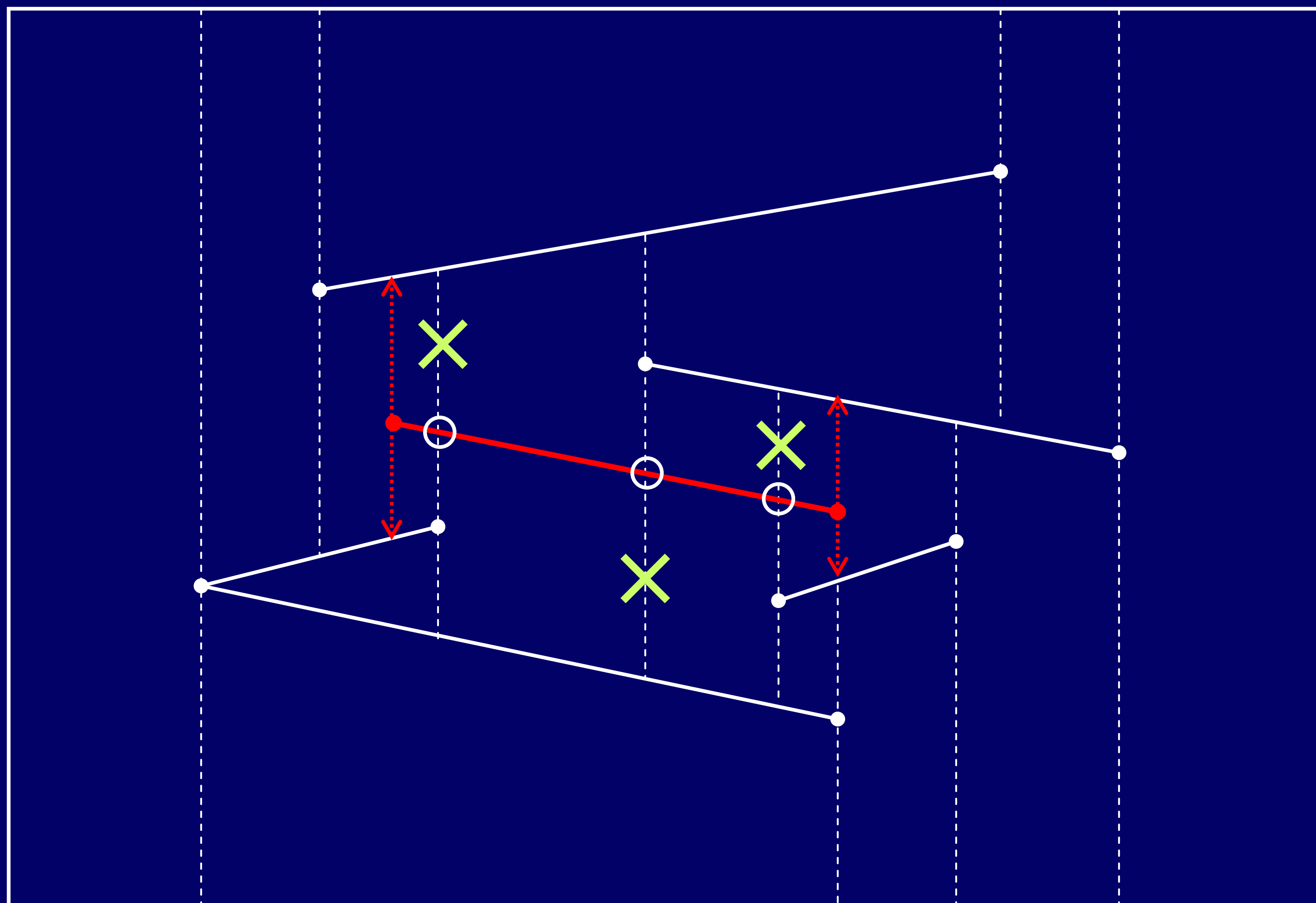
- Cada trapezóide é “definido” por 4 entidades
  - Segmento de cima
  - Segmento de baixo
  - Vértice da esquerda
  - Vértice da direita



Construção

- Algoritmo de varredura
  - Semelhante ao algoritmo de rasterização de polígonos e ao algoritmo para detecção de interseções entre segmentos de retas
- Algoritmo de inserção randomizada
  - Segmentos são embaralhados e inseridos um a um na estrutura
    - Localiza-se em qual trapézio a extremidade esquerda do segmento cai
    - Detecta-se quais “caminhos de bala” o segmento intersecta
    - Dispara-se balas para cima e para baixo a partir das extremidades

Construção



Construção

- Para localizar o trapézio onde a extremidade cai utiliza-se uma estrutura de busca
  - É neste aspecto que estamos interessados
  - Veremos mais tarde que esta operação pode ser feita em $O(\log n)$
- Para descobrir os trapézios intersectados pelo segmento e efetuar os disparos utiliza-se uma estrutura de dados própria para representar a topologia de subdivisões poligonais
  - *DCEL*
  - *Half-edge*
  - etc

Atualização da Estrutura

- Ignorando o aspecto da localização da extremidade esquerda, a inserção do pelo i'ésimo segmento tem complexidade $O(k_i)$, onde k_i é o número de trapézios criados
  - Um novo trapezóide é criado para cada "caminho de bala" intersectado
  - Cada um dos 4 disparos pode levar à criação de um novo trapezóide
  - Cada uma dessas operações pode ser feita em $O(1)$ usando uma estrutura de dados adequada

Análise

- **Lema:** O valor esperado de k_i é O (1)
  - Isto é, numa construção incremental randomizada, a inserção de um novo segmento no mapa gera na média um número constante de trapézios
  - Como conseqüência, cada inserção tem complexidade O (1 + log n) = O (log n)
    - O fator log n se refere à busca da extremidade esquerda do segmento
- A prova do lema é baseada na técnica de análise "para trás" (*backward*)

Prova do Lema

- Seja T_i o mapa trapezoidal logo após a inserção do i-ésimo segmento e S_i o conjunto de todos os segmentos de reta que compõem T_i
- Para estimarmos o número médio de trapézios criados pela inserção do i-ésimo segmento, precisamos considerar todas as possíveis permutações dos i segmentos
- Cada segmento $s \in S_i$ tem probabilidade $1/i$ de ter sido o último inserido
- Se denotarmos por $k_{i,s}$ o número de trapézios criados no caso de s ter sido o último, então

$$E[k_i] = \sum_{s \in S_i} \frac{1}{i} k_{i,s} = \frac{1}{i} \sum_{s \in S_i} k_{i,s}$$

Prova do Lema

- Vamos dizer que um trapézio $t \in T_i$ depende de um segmento s se t é criado em conseqüência de s ter sido inserido por último
- Definimos a função $\delta(t, s)$ como sendo 1 se t depende de s e 0 caso contrário. Então

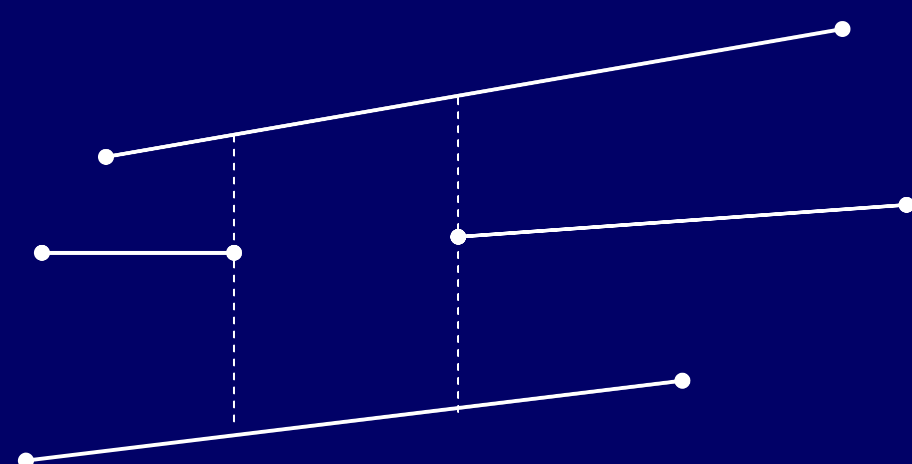
$$k_{i,s} = \sum_{t \in T_i} \delta(t,s)$$

- Portanto,

$$E[k_i] = \frac{1}{i} \sum_{s \in S_i} k_{i,s} = \frac{1}{i} \sum_{s \in S_i} \sum_{t \in T_i} \delta(t,s)$$

Prova do Lema

- A dificuldade reside em determinar quantos trapézios dependem de um dado segmento
  - Alguns segmentos podem gerar muitos trapézios enquanto que outros, poucos
- Entretanto, se perguntarmos **de quantos segmentos depende um trapézio**, vemos que a resposta é no máximo 4
  - (Trapézios degenerados em triângulos dependem de apenas 3)



Prova do Lema

- Portanto, podemos inverter a ordem dos somatórios

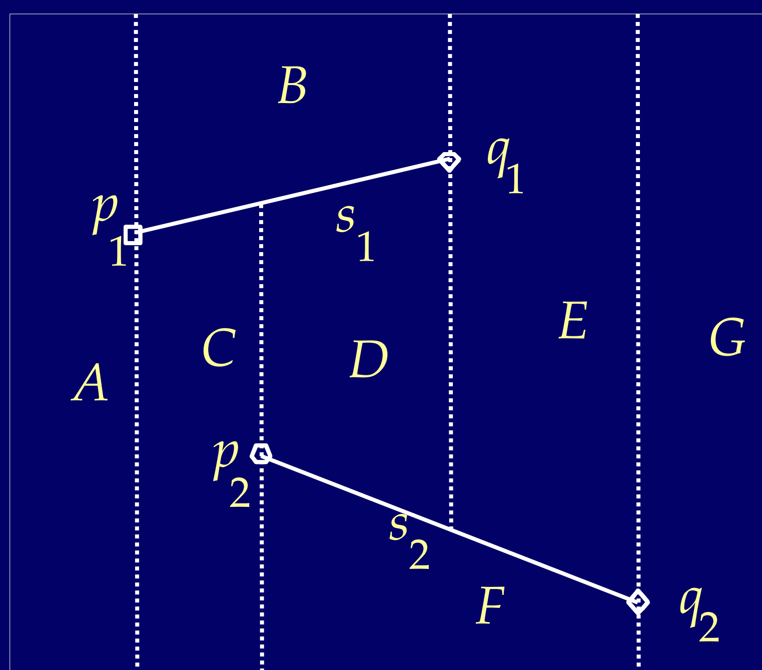
$$E[k_i] = \frac{1}{i} \sum_{s \in S_i} \sum_{t \in T_i} \delta(t,s)$$

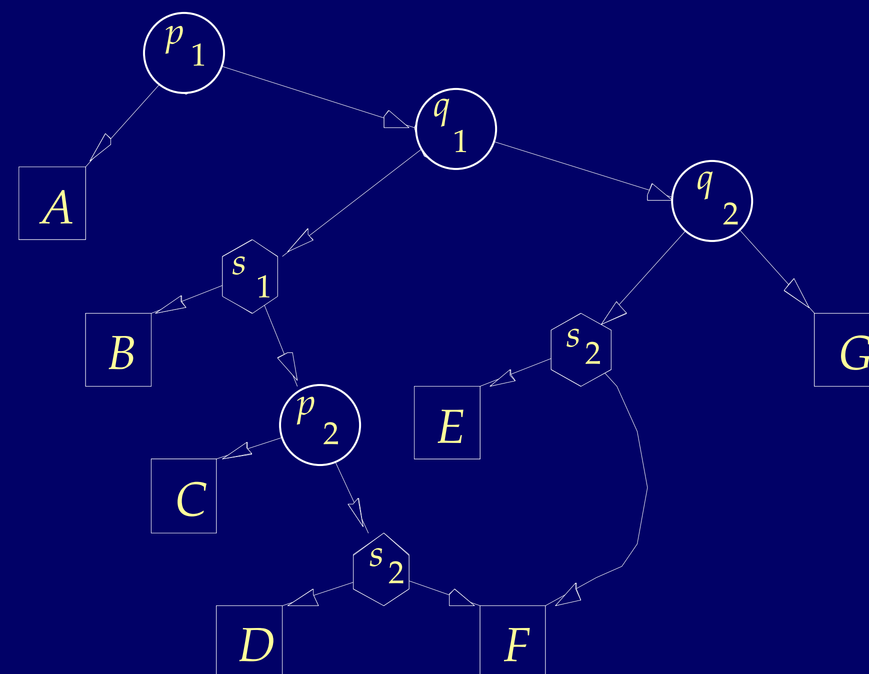
$$= \frac{1}{i} \sum_{t \in T_i} \sum_{s \in S_i} \delta(t,s) \leq \frac{1}{i} \sum_{t \in T_i} 4 = \frac{1}{i} 4 |T_i| = \frac{1}{i} 4 O(i) = O(1)$$

Estrutura de Busca

- É um grafo acíclico direcionado
  - Parece com uma árvore binária, mas há compartilhamento de sub-árvores
- Nós-folha representam os trapezóides
- Nós internos de dois tipos
  - <u>Nós x</u> representam coordenadas x de extremidades de segmentos de reta
    - Os nós à esquerda/direita têm coordenadas x menores/maiores que a do ponto extremo
  - <u>Nós y</u> contêm ponteiros para segmentos de reta
    - Os nós à esquerda/direita representam regiões acima/abaixo do segmento
    - Só visitamos um nó y se sabemos que o ponto procurado tem coordenada x entre as extremidades

Estrutura de Busca

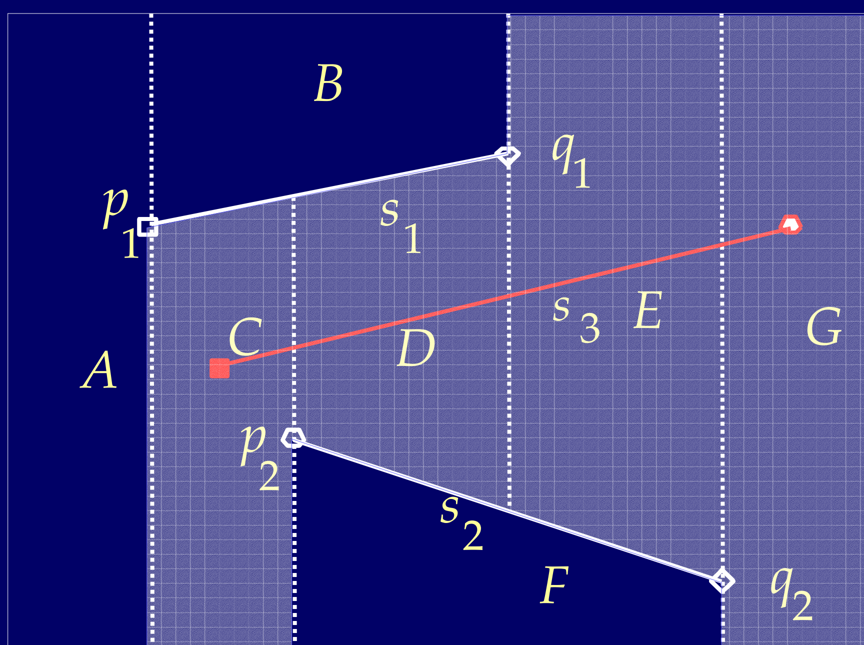


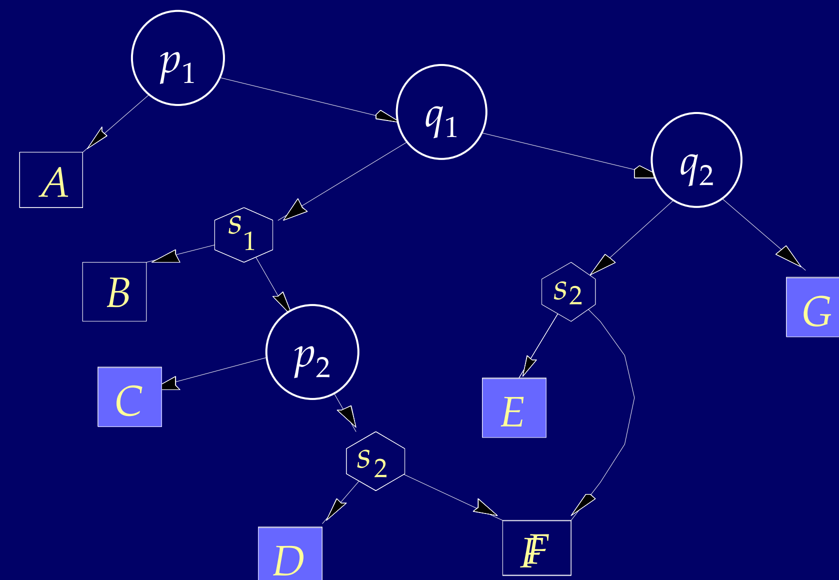


Construção da Estrutura de Busca

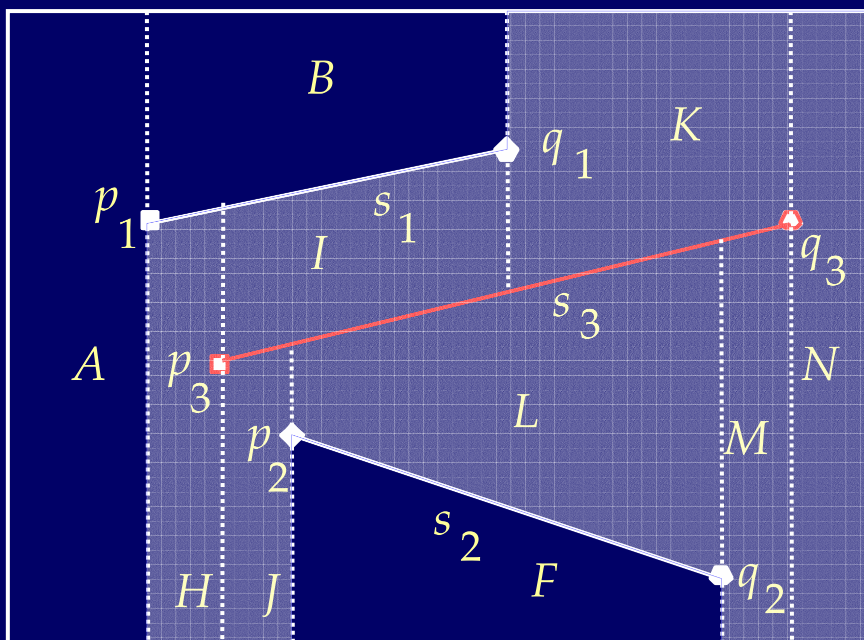
- A estrutura de busca é construída de forma incremental em paralelo com o mapa poligonal
  - Em qualquer dado instante a estrutura de busca pode ser usada para localizar um ponto no mapa poligonal
- A inserção de um novo segmento faz com que alguns trapezóides (folhas) sejam removidos e substituídos por outros
  - Na estrutura, são alteradas apenas os lugares onde essas folhas existiam

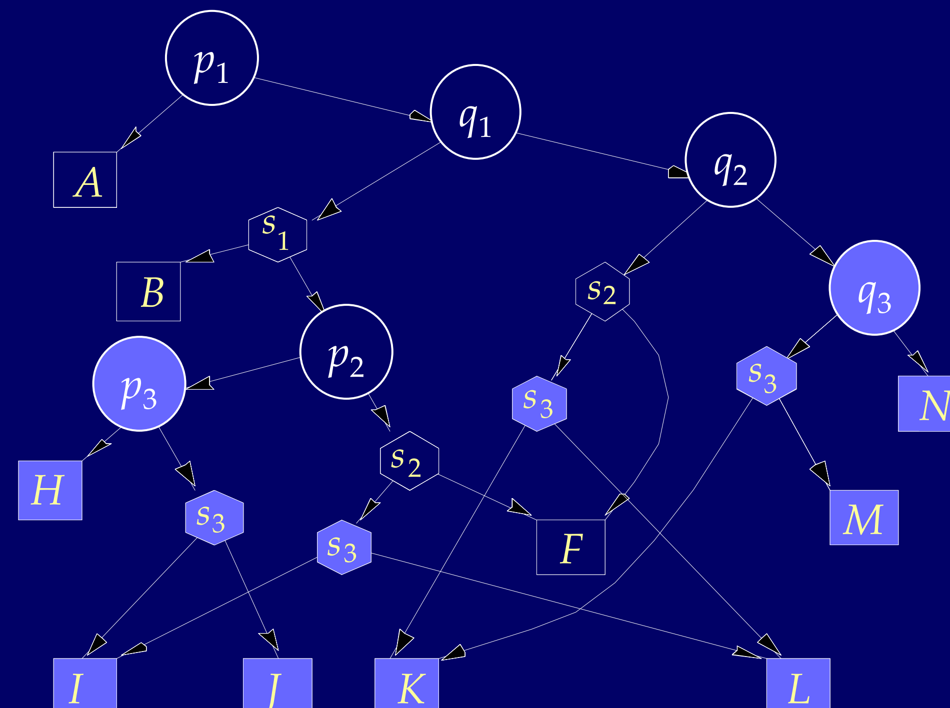
Construção da Estrutura de Busca





Construção da Estrutura de Busca





Construção da Estrutura de Busca

- Se o segmento inserido perpassa totalmente um trapezóide existente,
  - O local correspondente é substituído por uma sub-árvore com um nó y apontando para os dois novos trapézios criados
- Se uma extremidade do novo segmento trapezóide existente
  - Local substituído por um nó x referente à extremidade
    - De um lado, coloca-se um nó y correspondente à separação dos trapézios acima / abaixo
    - Do outro lado, coloca-se uma referência para o trapézio restante à esquerda / direita

Análise

- A estrutura tem complexidade esperada de espaço $O(n)$
  - Facilmente provado uma vez que sabemos que o número esperado de trapézios criados a cada inserção é $O(1)$
- A busca de um ponto tem complexidade esperada $O(\log n)$
  - Prova um pouco mais sutil
- Esses limites de complexidade média dependem apenas da ordem de inserção e não dos segmentos

Prova da complexidade de tempo

- Normalmente a prova de complexidade média consistiria em analisar pontos de consulta escolhidos randomicamente
- Ao invés disso, vamos admitir apenas um ponto de consulta q e o efeito de todas as possíveis ordens de inserção dos segmentos
  - Podemos pensar que q é escolhido por um "adversário" que tenta escolher o pior lugar possível mas não sabe em que ordem os segmentos serão inseridos

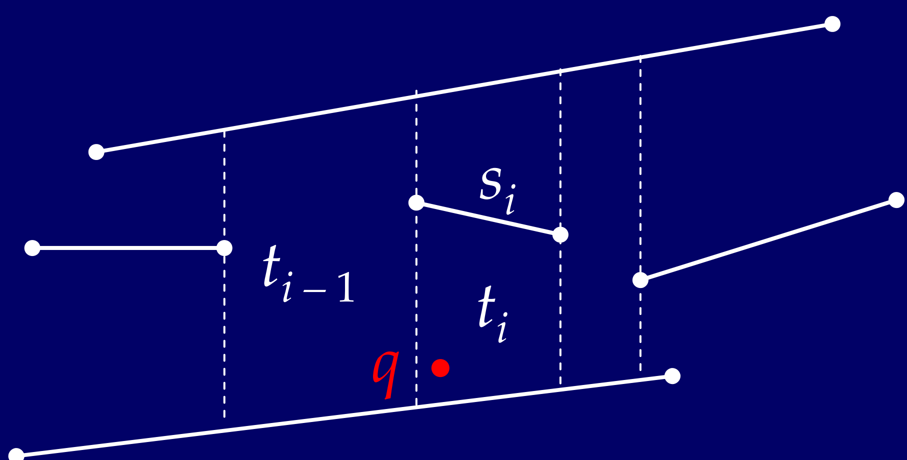
Prova da complexidade de tempo

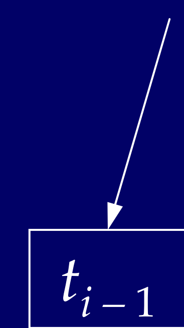
- A complexidade de uma consulta é dada pelo comprimento do caminho percorrido na estrutura de dados (número de nós)
- O caminho depende da ordem em que os segmentos foram inseridos
- Vamos analisar como q se move com respeito à estrutura à medida que cada novo segmento é inserido
- Vamos chamar de t_i o trapezóide dentro do qual q se encontra após a inserção do i-ésimo segmento

Prova da complexidade de tempo

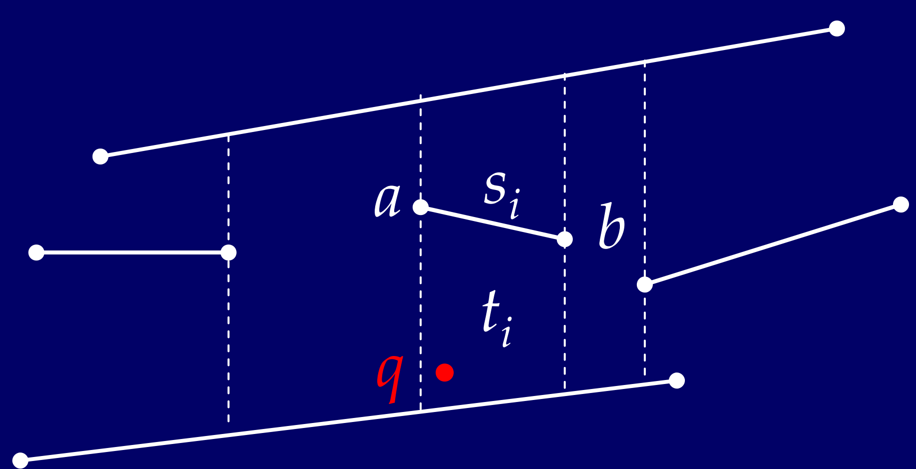
- Se $t_i = t_{i-1}$, então a inserção do i-ésimo segmento não afetou a estrutura nas imediações de q
  - Tempo de consulta após a i-ésima inserção permanece inalterado
- Se $t_i \neq t_{i-1}$, então a inserção do i-ésimo segmento causou a remoção de t_{i-1}
  - No pior caso, t_{i-1} foi substituído na estrutura por 4 novos trapézios, sendo t_i um deles
  - Caminho até t_i fica 3 nós mais compridos

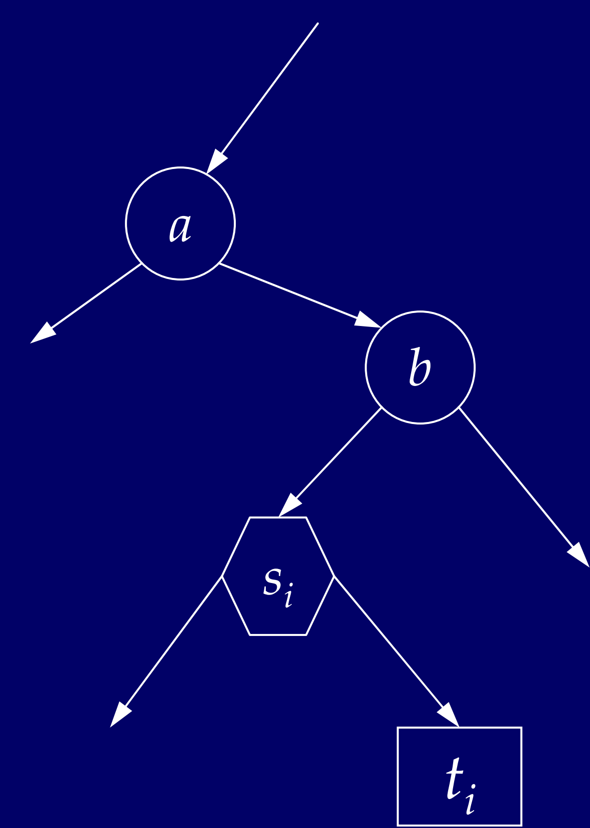
Prova da complexidade de tempo





Prova da complexidade de tempo





Prova da complexidade de tempo

- Qual a probabilidade P_i de que o trapezóide em que q se encontra tenha mudado após a i-ésima inserção (isto é, de que $t_i \neq t_{i-1}$)?
  - Observamos que t_i é delimitado por 4 segmentos
  - A probabilidade de um segmento ter sido o i-ésimo a ser inserido é $1/i$
  - Conseqüentemente, $P_i = 4/i$
- O comprimento do caminho desde a raiz até t_i é portanto

$$\sum_{i=1}^{n} P_i = \sum_{i=1}^{n} 3\frac{4}{i} = 12\sum_{i=1}^{n}\frac{1}{i} \approx 12\ln n = O(\log n)$$

Um resultado mais forte

- Embora o tempo de busca médio seja $O(\log n)$, é possível haver caminhos com comprimentos longos
  - Consultas repetidas nesses caminhos podem comprometer o desempenho da estrutura
- Existe um resultado mais forte a respeito da altura da estrutura
  - Lema é provado no livro

Um resultado mais forte

- **Lema:** Dado um conjunto disjunto de n segmentos de reta e um parâmetro $\lambda>0$, a probabilidade de que a estrutura tenha altura maior que $3\ \lambda \ln (n+1)$ é de no máximo $2 / (n+1)^{\lambda \ln 1.25 - 3}$
  - Por exemplo, para $\lambda = 20$, a probabilidade de que a altura da estrutura seja maior que $60 \ln (n+1)$ é de aproximadamente $2 / (n+1)^{1.25}$

Garantindo busca logarítmica no pior caso

- O lema diz que, à medida que n aumenta, a probabilidade de que a altura máxima seja não logarítmica aumenta ainda mais rápido
- Isto sugere que podemos tentar construir a estrutura várias vezes (com permutações randômicas de inserção) até termos uma estrutura com altura aceitável
- O número esperado de tentativas varia com a meta que estipulamos
  - Para uma meta de altura suficientemente grande, a probabilidade de a alcançarmos será constante