Dualidad Punto-Línea y Convex Hull Trick

Agustín Santiago Gutiérrez

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Training Camp Argentina 2018

Contenidos

Advertencia: Estas diapos no son autocontenidas. Les faltan todos los dibujos (pizarrón), y explicaciones clave.

“[...] geometry, however, is not concerned with the relation of the ideas involved in it to objects of experience, but only with the logical **connection of these ideas among themselves**.”

Albert Einstein, *Relativity: The special and general theory*, 1920

Contenidos

Cápsula convexa

- La forma que adopta una banda elástica, si la hacemos rodear "postes" clavados en los puntos.
- Se puede calcular con Graham Scan, en un solo recorrido.
- Se puede calcular en dos partes: La **cápsula superior** y la **cápsula inferior**.
- Ambos algoritmos se basan en "tener un stack e ir sacando hasta que encaje".

Contenidos

Problemas de ejemplo

- Cantidad de rectas “sobre las que pega la lluvia”
- Ancho de un conjunto de puntos en una cierta dirección
- Máxima cantidad de rectas concurrentes
- Cantidad de vértices de la cápsula convexa de puntos en el plano
- Máxima cantidad de puntos alineados
- “A qué altura pega la lluvia para un cierto x”
- Máxima cantidad de “gráficos de rectas incomparables”
- Mínima cantidad de torres necesarias para apilar mesadas

Problemas de ejemplo

- Cantidad de rectas “sobre las que pega la lluvia”
- Ancho de un conjunto de puntos en una cierta dirección
- Máxima cantidad de rectas concurrentes
- Cantidad de vértices de la cápsula convexa de puntos en el plano
- Máxima cantidad de puntos alineados
- “A qué altura pega la lluvia para un cierto x”
- Máxima cantidad de “gráficos de rectas incomparables”
- Mínima cantidad de torres necesarias para apilar mesadas

¿Cuáles de estos se relacionan?

Parejas de problemas duales

- Máxima cantidad de puntos alineados
- Máxima cantidad de rectas concurrentes

- Mínima cantidad de torres necesarias para apilar mesadas
- Máxima cantidad de “gráficos de rectas incomparables”

- Cantidad de vértices de la cápsula convexa de puntos en el plano
- Cantidad de rectas “sobre las que pega la lluvia”

- “A qué altura pega la lluvia para un cierto x”
- Ancho de un conjunto de puntos en una cierta dirección

Parejas de problemas duales

- Máxima cantidad de puntos alineados
- Máxima cantidad de rectas concurrentes

- Mínima cantidad de torres necesarias para apilar mesadas
- Máxima cantidad de “gráficos de rectas incomparables”

- Cantidad de vértices de la cápsula convexa de puntos en el plano
- Cantidad de rectas “sobre las que pega la lluvia”

- “A qué altura pega la lluvia para un cierto *x*”
- Ancho de un conjunto de puntos en una cierta dirección

¿Les parecen igual de fáciles/difíciles los emparejados?

Contenidos

Representación de rectas en computadora

- Idea: ¿Cómo representamos una recta?

Representación de rectas en computadora

- Idea: ¿Cómo representamos una recta?
- Dependiendo del contexto y de cómo la vamos a usar hay muchas maneras.
- Para “cálculos geométricos con rectas y figuras”, en general representaciones con **vectores** es lo mejor.

Representación de rectas en computadora

- Idea: ¿Cómo representamos una recta?
- Dependiendo del contexto y de cómo la vamos a usar hay muchas maneras.
- Para “cálculos geométricos con rectas y figuras”, en general representaciones con **vectores** es lo mejor.
- Si la recta representa naturalmente una cierta *función lineal* que a veces evaluamos, la **forma explícita** $y = ax + b$ puede ser más útil.
- Notar que de ser así no tienen sentido las rectas verticales (no son funciones).

Rectas y puntos en struct

```
struct Recta
{
    int a,b; // Forma explicita:
             // contiene los puntos (x,y)
             // que cumplen y = ax + b
};

struct Punto
{
    int x,y;
};
```

Rectas y puntos en struct

```
struct Recta
{
    int a,b; // Forma explicita:
             // contiene los puntos (x,y)
             // que cumplen y = ax + b
};

struct Punto
{
    int x,y;
};
```

- ¿Qué diferencia hay entre ambas?

Rectas y puntos en struct

```
struct Recta
{
    int a,b; // Forma explicita:
             // contiene los puntos (x,y)
             // que cumplen y = ax + b
};

struct Punto
{
    int x,y;
};
```

- ¿Qué diferencia hay entre ambas?
- Los nombres. “Names Don’t Constitute Knowledge” R. Feynman https://www.youtube.com/watch?v=lFIYKmos3-s

La transformación

- Transformar rectas en puntos: $y = ax + b \mapsto (a, b)$

La transformación

- Transformar rectas en puntos: $y = ax + b \mapsto (a, b)$
- ¿Eso es todo? ¿Eso es dualidad punto recta?

La transformación

- Transformar rectas en puntos: $y = ax + b \mapsto (a, b)$
- ¿Eso es todo? ¿Eso es dualidad punto recta?
- Sí.

La transformación

- Transformar rectas en puntos: $y = ax + b \mapsto (a, b)$
- ¿Eso es todo? ¿Eso es dualidad punto recta?
- Sí.
- ¿De verdad?

La transformación

- Transformar rectas en puntos: $y = ax + b \mapsto (a, b)$
- ¿Eso es todo? ¿Eso es dualidad punto recta?
- Sí.
- ¿De verdad?
- Sí.

La transformación

- Transformar rectas en puntos: $y = ax + b \mapsto (a, b)$
- ¿Eso es todo? ¿Eso es dualidad punto recta?
- Sí.
- ¿De verdad?
- Sí.
- Pero para que sea útil hay que entender la **relación** entre ellos.

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$
- "rectas no verticales" = " funciones lineales" $\Leftrightarrow$ "puntos del plano"

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$
- “rectas no verticales” = “ funciones lineales” $\Leftrightarrow$ “puntos del plano”
- “pendiente mayor” $\Leftrightarrow$ “coordenada x mayor”

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$
- “rectas no verticales” = “ funciones lineales” $\Leftrightarrow$ “puntos del plano”
- “pendiente mayor” $\Leftrightarrow$ “coordenada x mayor”
- “rectas paralelas” $\Leftrightarrow$ “alineados en vertical”

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$
- "rectas no verticales" = " funciones lineales" $\Leftrightarrow$ "puntos del plano"
- "pendiente mayor" $\Leftrightarrow$ "coordenada x mayor"
- "rectas paralelas" $\Leftrightarrow$ "alineados en vertical"
- "rectas concurrentes" $\Leftrightarrow$ "alineados salvo verticales"

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$
- "rectas no verticales" = " funciones lineales" $\Leftrightarrow$ "puntos del plano"
- "pendiente mayor" $\Leftrightarrow$ "coordenada x mayor"
- "rectas paralelas" $\Leftrightarrow$ "alineados en vertical"
- "rectas concurrentes" $\Leftrightarrow$ "alineados salvo verticales"
- "y" (o sea $f(x)$ dado x) $\Leftrightarrow p \cdot (x, 1)$

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$
- “rectas no verticales” = “ funciones lineales” $\Leftrightarrow$ “puntos del plano”
- “pendiente mayor” $\Leftrightarrow$ “coordenada x mayor”
- “rectas paralelas” $\Leftrightarrow$ “alineados en vertical”
- “rectas concurrentes” $\Leftrightarrow$ “alineados salvo verticales”
- “y” (o sea $f(x)$ dado x) $\Leftrightarrow p \cdot (x, 1)$
- “Mayor $f(x)$” $\Leftrightarrow$ “p está más avanzado en la dirección apuntada por $\vec{v} = (x, 1)$”.

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$
- “rectas no verticales” = “ funciones lineales” $\Leftrightarrow$ “puntos del plano”
- “pendiente mayor” $\Leftrightarrow$ “coordenada x mayor”
- “rectas paralelas” $\Leftrightarrow$ “alineados en vertical”
- “rectas concurrentes” $\Leftrightarrow$ “alineados salvo verticales”
- “y” (o sea $f(x)$ dado x) $\Leftrightarrow$ $p \cdot (x, 1)$
- “Mayor $f(x)$” $\Leftrightarrow$ “p está más avanzado en la dirección apuntada por $\vec{v} = (x, 1)$”.
- **“Rectas sobre las que pega la lluvia” $\Leftrightarrow$ “Cápsula superior”.**

Relaciones

- $y = f(x) = ax + b \Leftrightarrow (a, b) = p$
- "rectas no verticales" = " funciones lineales" $\Leftrightarrow$ "puntos del plano"
- "pendiente mayor" $\Leftrightarrow$ "coordenada x mayor"
- "rectas paralelas" $\Leftrightarrow$ "alineados en vertical"
- "rectas concurrentes" $\Leftrightarrow$ "alineados salvo verticales"
- "y" (o sea $f(x)$ dado x) $\Leftrightarrow p \cdot (x, 1)$
- "Mayor $f(x)$" $\Leftrightarrow$ "p está más avanzado en la dirección apuntada por $\vec{v} = (x, 1)$".
- **"Rectas sobre las que pega la lluvia" $\Leftrightarrow$ "Cápsula superior".**
- Esta última es la que da el nombre a "convex hull trick"
- ¡Revisar los problemas anteriores para verificar los emparejamientos!

Contenidos

Qué es

- Es la técnica que usamos para calcular eficientemente,a partir de un x, el valor de $max_{i=1}^{n} f_i(x)$, si tenemos n funciones lineales f_i

Qué es

- Es la técnica que usamos para calcular eficientemente,a partir de un x, el valor de $max_{i=1}^{n} f_i(x)$, si tenemos n funciones lineales f_i
- Es como tener las “Rectas donde pega la lluvia” rápido, que son las que importan. O por dualidad, es como tener la cápsula superior, que son los únicos puntos candidatos a ser los más alejados en una dirección $\vec{v} = (x, 1)$.

Convex Hull Trick offline

- Si las rectas van apareciendo **en orden de pendiente**, o si las podemos ordenar libremente, podemos hacer como al construir la cápsula superior: vamos “descartando hasta que encaje” del vector de rectas, y siempre tendremos las rectas de lluvia **ordenadas** (binary search).

Convex Hull Trick online

- Si las rectas aparecen en forma dinámica y en cualquier orden, podemos tenerlas en un `set` **ordenadas por pendiente**, y usar `lower_bound` para ver dónde insertar la nueva (si es que hay que hacerlo).

Convex Hull Trick online

- Si las rectas aparecen en forma dinámica y en cualquier orden, podemos tenerlas en un `set` **ordenadas por pendiente**, y usar `lower_bound` para ver dónde insertar la nueva (si es que hay que hacerlo).
- Podemos usar la dualidad para razonar más fácil en términos de puntos (si así nos resulta más fácil), pues es análogo a mantener la cápsula superior cuando aparecen nuevos puntos: allí los mantendríamos ordenados por coordenada x, y luego "borraríamos adyacentes a cada lado" hasta que la banda elástica quede "correctamente tensada".

Convex Hull Trick online

- Si las rectas aparecen en forma dinámica y en cualquier orden, podemos tenerlas en un `set` **ordenadas por pendiente**, y usar `lower_bound` para ver dónde insertar la nueva (si es que hay que hacerlo).
- Podemos usar la dualidad para razonar más fácil en términos de puntos (si así nos resulta más fácil), pues es análogo a mantener la cápsula superior cuando aparecen nuevos puntos: allí los mantendríamos ordenados por coordenada x, y luego "borraríamos adyacentes a cada lado" hasta que la banda elástica quede "correctamente tensada".
- Truco de implementación: suele ser conveniente agregar rectas centinelas en los extremos, con pendientes enormes, para evitar problemas por accesos fuera de rango.
- Equivalentemente, para trabajar con la cápsula superior de un conjunto de puntos, agregaríamos dos puntos centinela con valores extremos de la coordenada x.

Contenidos

Otro ejemplo de transformación

- Consiste en pensar un intervalo (A, B) en la recta numérica, como un punto (A, B) en el plano.
- La recta numérica original puede pensarse como la recta $y = x$, que contiene los intervalos de longitud 0 (puntos).
- Todos los intervalos válidos viven en la parte superior $y > x$ del plano.
- Los (A, B) con $A > B$ podrían usarse para representar el “complemento” de un intervalo: tiene sentido en problemas circulares.

Relaciones

- Un intervalo contiene a otro, si y solo si este último está más “abajo a la derecha”.
- Si imaginamos que el punto de un intervalo lanza una “sombra” haba abajo y hacia la derecha, los puntos de la recta $y = x$ cubiertos son exactamente el intervalo original.
- Así que dos intervalos se tocan si y solo si sus sombras chocan.
- Dado un punto (que representa un intervalo), los puntos de intervalos que se intersecan con él, son los que están en un “cuadrante” arriba y a la izquierda de su reflejo por $y = x$.
- Para algunos es más fácil visualizar puntos que visualizar intervalos, y la transformación puede facilitar tener algunas ideas clave.

Problema de ejemplo

- Updates: Aparecen y desaparecen puntos con peso. Todos en la parte **superior** $y > x$ del plano.
- Query: Dado un punto del plano en la parte **inferior** $y < x$, suma de los pesos de los que están arriba y a su izquierda.

Problema de ejemplo

- Updates: Aparecen y desaparecen puntos con peso. Todos en la parte **superior** $y > x$ del plano.
- Query: Dado un punto del plano en la parte **inferior** $y < x$, suma de los pesos de los que están arriba y a su izquierda.
- Este problema es el equivalente exacto que resulta al aplicar la dualidad intervalo-punto que vimos, al problema F del TAP 2017