Geometría

Leopoldo Taravilse   Francisco Roslán

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Training Camp 2012

Contenidos

Contenidos

Problema

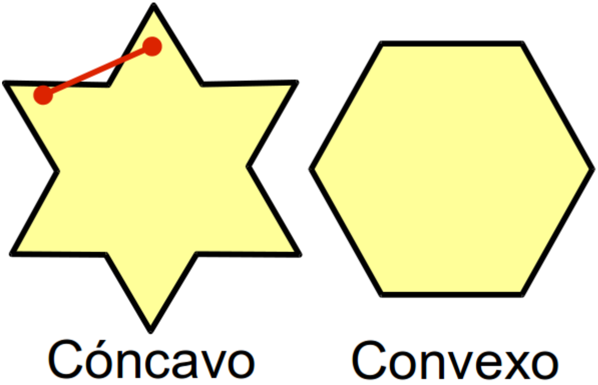
Problema: Dado un polígono, decidir si hay un punto en el polígono que no es visible desde todo el polígono.

Problema

Problema: Dado un polígono, decidir si hay un punto en el polígono que no es visible desde todo el polígono.
Esto es equivalente a preguntar si el polígono es cóncavo o convexo

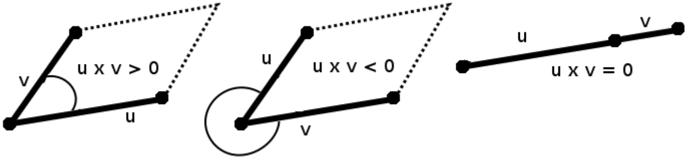
Problema

Problema: Dado un polígono, decidir si hay un punto en el polígono que no es visible desde todo el polígono.
Esto es equivalente a preguntar si el polígono es cóncavo o convexo



Producto cruz

- Producto cruz $u \times v$
- $u = (x_1, y_1)$, $v = (x_2, y_2)$, $u \times v = x_1 * y_2 - x_2 * y_1 = |u||v|\sin(\phi)$
- El valor absoluto es el área del paralelogramo.
- El signo es la orientación de los vectores, del ángulo ϕ.



Algoritmo $O(n)$

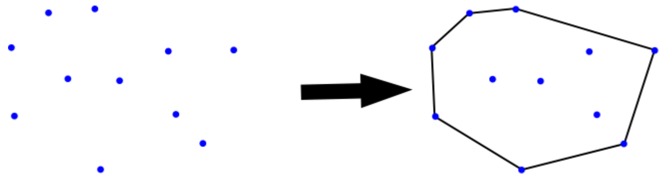
```
bool isConvex(vector<int> x, vector<int> y)
{
    int n = x.size(), pos = 0, neg = 0;
    for(int i = 0; i < n; i++)
    {
        int j = (i + n - 1) %n, k = (i + 1) %n;
        int pc = (x[k]-x[i])*(y[j]-y[i])-(x[j]-x[i])*(y[k]-y[i]);
        if(pc < 0)
            neg++;
        else
            pos++;
    }
    return (neg == 0) || (pos == 0);
}
```

Contenidos

Cápsula Convexa

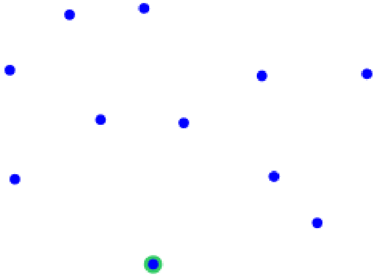
Definición

Dados n puntos, la cápsula convexa es el subconjunto de esos puntos que forma un polígono convexo y que contiene a todos los demás puntos en su interior. Se puede probar que es única.



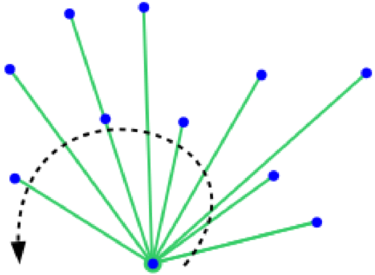
Algoritmo de Graham

El punto de más abajo, y si hay que desempatar, el de más a la izquierda (igual se puede rotar para que haya uno sólo más abajo que los demás) seguro que está en la cápsula convexa. Encontrar este punto tiene complejidad $O(n)$.



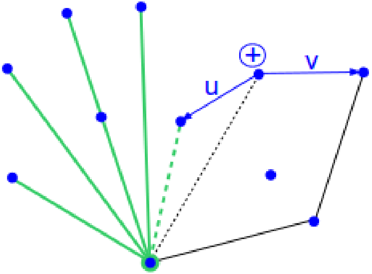
Algoritmo de Graham

Ordenamos los puntos por ángulo, en sentido anti-horario, y para desempatar el más cercano primero. Esto toma $O(n \log n)$.



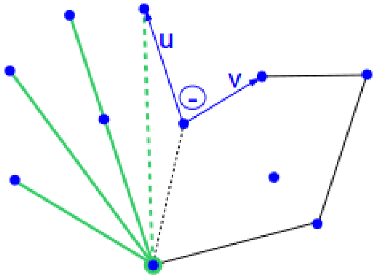
Algoritmo de Graham

El próximo punto siempre está en la Convex Hull. En este caso $u \times v > 0$.



Algoritmo de Graham

Mientras $u \times v < 0$ sacamos el punto anterior. Este paso toma $O(n)$



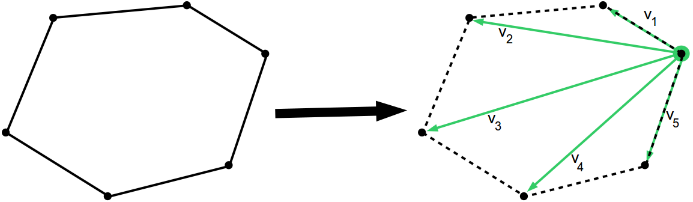
Problemas

- goo.gl/rT7Ji
- goo.gl/lIEHC

Contenidos

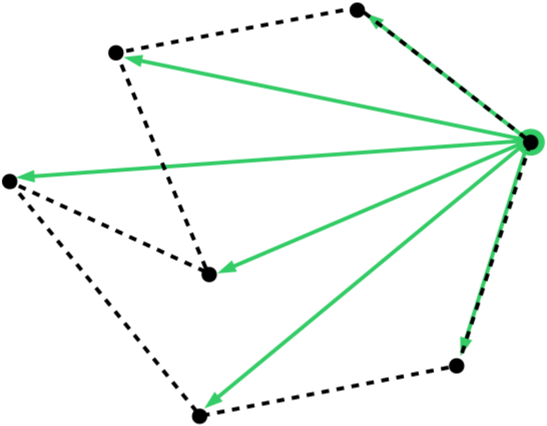
Superficie de un polígono

Si el polígono es convexo, se triangula desde un vértice cualquiera y se suman en orden los productos cruz. La superficie del polígono es la mitad del valor absoluto de este número.



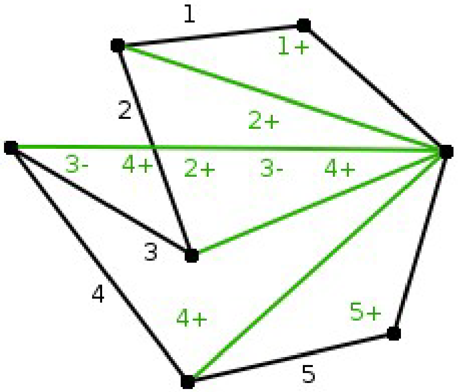
Superficie de un polígono

¿Y qué pasa si el polígono es cóncavo?



Superficie de un polígono

Es lo mismo.

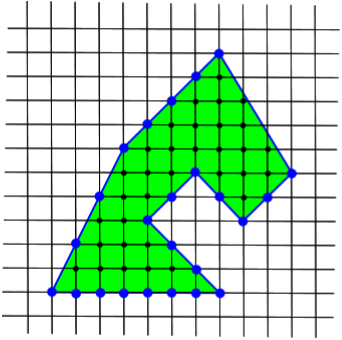


Contenidos

Teorema de Pick

Teorema de Pick:

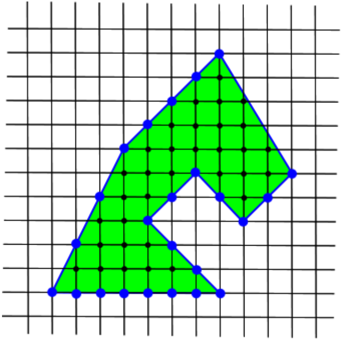
- $A = I + \frac{B}{2} - 1$
- $I = A - \frac{B}{2} + 1$



Teorema de Pick

Teorema de Pick:

- A es el área del polígono.
- I es la cantidad de puntos interiores.
- B es la cantidad de puntos del borde. Se pueden calcular usando el MCD de la longitud en x y en y de cada lado.



Para practicar

- goo.gl/Lw3HU

Contenidos

Divide & Conquer

Divide and Conquer es una técnica que consiste en dividir a un problema en subproblemas más chicos, resolver los problemas más chicos de la misma manera que el original hasta llegar a un caso base, y luego unir las soluciones de los problemas más chicos para formar la solución del problema original.

Divide & Conquer

Divide and Conquer es una técnica que consiste en dividir a un problema en subproblemas más chicos, resolver los problemas más chicos de la misma manera que el original hasta llegar a un caso base, y luego unir las soluciones de los problemas más chicos para formar la solución del problema original.
La diferencia con programación dinámica es que por lo general los subproblemas no se relacionan entre sí (no se reutiliza información).
Igualmente podemos decir que es una técnica recursiva.
Un ejemplo de algoritmo que utiliza la técnica de Divide and Conquer es el algoritmo de ordenamiento merge sort.

Merge Sort

- Si recibimos una lista de n números y $n \geq 2$ la dividimos en dos partes tal que una tenga a lo sumo un elemento más que la otra.

Merge Sort

- Si recibimos una lista de n números y $n \geq 2$ la dividimos en dos partes tal que una tenga a lo sumo un elemento más que la otra.
- Llamamos a Merge Sort recursivamente para que ordene las dos partes.

Merge Sort

- Si recibimos una lista de n números y $n \geq 2$ la dividimos en dos partes tal que una tenga a lo sumo un elemento más que la otra.
- Llamamos a Merge Sort recursivamente para que ordene las dos partes.
- Una vez que tenemos las dos mitades ordenadas recorremos ambas mitades con dos punteros eligiendo siempre el elemento más chico y avanzando el puntero en esa lista.

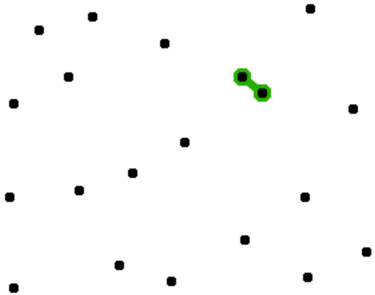
Merge Sort

- Si recibimos una lista de n números y $n \geq 2$ la dividimos en dos partes tal que una tenga a lo sumo un elemento más que la otra.
- Llamamos a Merge Sort recursivamente para que ordene las dos partes.
- Una vez que tenemos las dos mitades ordenadas recorremos ambas mitades con dos punteros eligiendo siempre el elemento más chico y avanzando el puntero en esa lista.
- La complejidad de este algoritmo es $O(n \log n)$. La forma más fácil de probar esto es asumir que n es potencia de 2 y ver luego que si $m < n$ el algoritmo termina más rápido para m que para n.

Contenidos

Par de puntos más cercanos

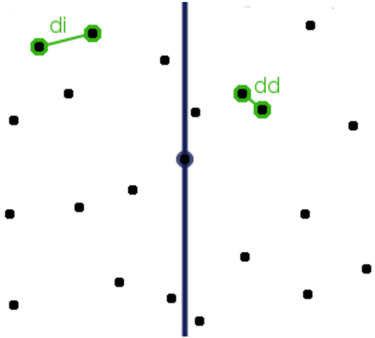
Dados *n* puntos, ¿cuál es el par de puntos que están más cerca entre sí que cualquier otro par entre los *n* puntos?



¿Qué tiene que ver Divide and Conquer con geometría? Este problema sale con un algoritmo de Divide and Conquer.

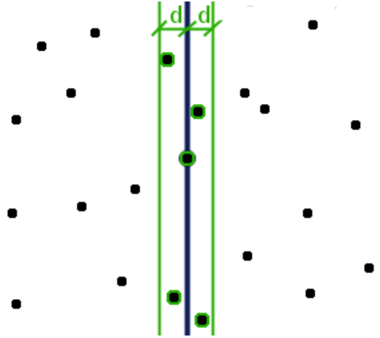
Par de puntos más cercanos

Dividimos los puntos en la mitad por coordenada x y resolvemos el problema recursivamente para cada mitad.



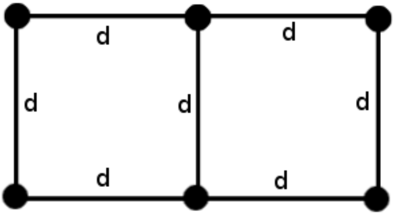
Par de puntos más cercanos

Ahora tenemos que calcular el par de puntos más cercanos tal que están uno de cada lado de la recta que divide a los puntos en dos mitades, y tomar mínimo.



Par de puntos más cercanos

Si d es el mínimo de las soluciones para los dos problemas, tomamos los puntos que están a distancia a lo sumo d de la recta del medio y los ordenamos verticalmente, luego cada punto puede tener a lo sumo 6 puntos a distancia menor o igual a d verticalmente. Para calcular el nuevo d hacemos esas 6 comparaciones.



 Par de puntos más cercanos

Fin de la clase