UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ciencias

Autor: Adrián Aguilera Moreno

Geometría Computacional

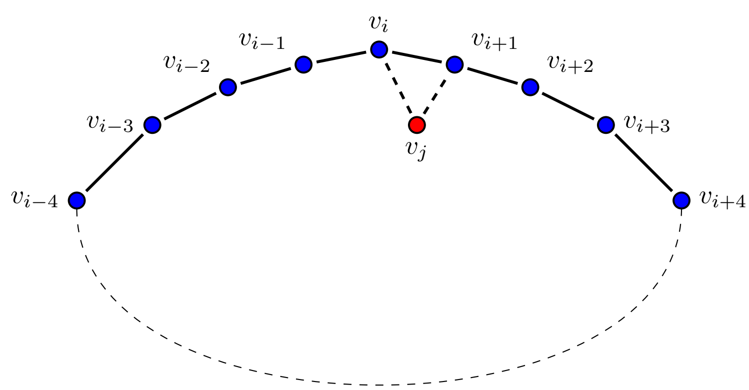
Tarea 01

1. Sea S un conjunto de puntos en el plano en posición general. Demuestra que el cierre convexo de S es el polígono convexo con perímetro y área más pequeños que contienen a S.

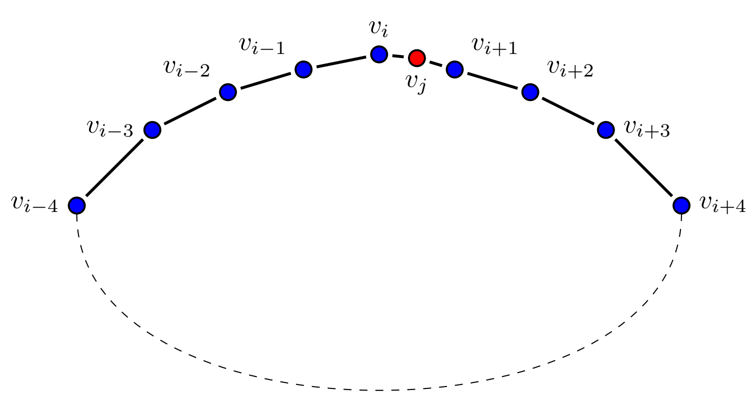
Dem. Para este problema, dividamos la prueba en dos posibles casos:

- Con $C(S)$[1] el cierre convexo. $C(S)$ es el polígono convexo con perímetro más pequeño que contiene a S.
  Prueba por *reducción al absurdo.* Supongamos que $C(S)$ no es el polígono de perímetro mínimo que envuelve todos los puntos en S. La suposición anterior implica que existe $C(S)' \neq C(S)$ tal que $P(C(S)') < P(C(S))$[2] con $C(S)'$ un segundo cierre convexo. Entonces, ¿Cómo modificar $C(S)$ para convertirlo en $C(S)'$? Esto puede ocurrir en 3 casos, estos son:

  Caso 1. Existe un punto interno al polígono que lo convierte en uno de menor perímetro, en este caso perdemos convexidad en la nueva figura y por tanto debemos descartar este caso (pues esta condición es parte del antecedente de nuestra implicación).



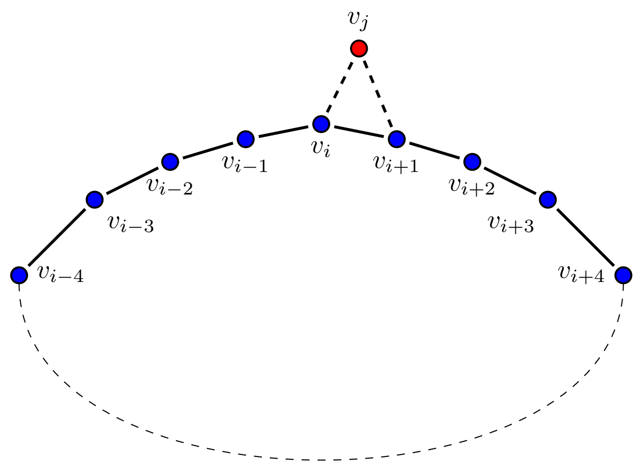
  Caso 2. Existe un punto, extra en $C(S)'$ y que no esta en $C(S)$, sobre un segmento de $C(S)$, esto no reduce el perímetro y por tanto podemos descartar este caso.



[1] Diremos que C es la función que nos devuelve como imagen el cierre convexo del conjunto de puntos que se pase como argumento.

[2] Diremos que P es la función que se mapea al perímitro del polígono que se pasa como argumento.

Caso 3. Este caso lo anexo solo para estar completo, pero no debe suceder por la definición de $C(S)$, pues es el polígono convexo formado por la envolvente convexa del conjunto de puntos en S, así no debe haber un punto que no quede interno de $C(S)$ y en consecuencia de $C(S)'$.



Por lo anterior, hemos llegado a una contradicción en cada caso exhibido. Pues ninguno cumple ser un polígono resultado de la envolvente convexa de menor perímetro que $C(S)$.

$$\therefore \text{ C(S) es de menor perímetro entre los que contienen a } S.$$

- Con $C(S)$ el cierre convexo. $C(S)$ es el polígono convexo con área más pequeña, tal que $C(S)$ contiene a S.
  Prueba por *reducción al absurdo*. Analicemos los casos 1 y 2, anteriores y supongamos que existe $A(C(S)') < A(C(S))$.
  - *Caso 1.* Perdemos convexidad y por tanto llegamos a una contradicción, pues cualquier punto interno nos trae como consecuencia la pérdida de convexidad.
  - *Caso 2.* El tener un punto en un segmento no disminuye el área del polígono, por tanto contradice que $A(C(S)') < A(C(S))$.

  Por lo anterior, hemos llegado a una contradicción en cada caso exhibido. Pues ninguno cumple ser un polígono resultado de la envolvente convexa de menor área que $C(S)$.

$$\therefore \text{ C(S) es de menor área entre los que contienen a } S.$$

2. Se define el diámetro de un conjunto de puntos S como la distancia más grande entre cualesquiera dos puntos de S, denotado por d. Demuestra que d está formado por dos vértices del cierre convexo de S.

Dem. Procedamos por reducción al absurdo.

Supongamos que el diámetro $C(S)$ no contiene dos puntos del diámetro, entonces, para $d = xy$;

- Los dos puntos de d están dentro de $C(S)$.

  Esto implica que d no es la distancia más grande entre cualesquiera dos puntos de S, pues tomando xv donde v es parte de la frontera de $C(S)$ tenemos que $||xv|| > ||xy||$. He aquí una contradicción de suponer que x, y están dentro de $C(S)$.

- Los dos puntos de d están fuera de $C(S)$.

  Esto implicaría que $C(S)$ no es el cierre convexo[3] de S y por tanto llegamos a una contradicción.

- O hay un punto de d dentro (x) y otro fuera (y) de $C(S)$.

  Esto implicaría que $C(S)$ no es el cierre convexo de S y por tanto hemos llegado a una contradicción.

Los casos anterior muestran que d esta formado por dos vértices del cierre convexo. □

3. Dado un conjunto de puntos S diseña un algoritmo que encuentre un polígono simple cuyos vértices sean el conjunto S.

Solución. Para el diseño de este algoritmo tomaremos como base el algoritmo `Graham` visto en clase, así, exhibamos el algoritmo

1. Primero debemos encontrar un punto distinguido p_0. Este punto puede ser cualquier punto y aquí nos podemos preguntar ¿Por qué cualquier punto? La respuesta es simple, todos los puntos serán parte de nuestro polígono y por tanto no da exactamente igual con cuál iniciar. Este paso tiene una complejidad contenida en $\mathcal{O}(1)$.

2. Ahora obtengamos un orden angular respecto a p_0, esto nos toma $\mathcal{O}(n \log_2 n)$ por la cota de ordenamiento por comparaciones existente.

3. Ahora que ya tenemos un orden angular y un punto por el cuál iniciar, recorramos nuestros puntos tomando como siguiente, siempre, al próximo en el orden (así creamos una arista en cada iteración en el recorrido y lo guardamos, digamos en una lista) en sentido contrario a las manecillas del reloj[4]. Esto nos simplifica el detalle de conocer nuestras direcciones (validar) para poder regresar en caso de un giro en sentido contrario al requerido en `Graham`. Esto nos toma la cantidad de puntos en S, por tanto tenemos un orden $\mathcal{O}(n)$. Eventualmente llegaremos a p_0 y es en este momento que nuestro algoritmo termina regresando las aristas encontradas durante el recorrido.

¿Por qué será cierto que nuestro algoritmo no encuentra aristas que se intersecten? Por el orden encontrado en (2).

Análisis de complejidad. nuestro algoritmo tiene una complejidad en

$$\therefore \mathcal{O}(1) + \mathcal{O}(n \log_2 n) + \mathcal{O}(n) \in \mathcal{O}(n \log_2 n).$$

[3]Pues y o x queda fuere de $C(S)$ y contradice la definición de C.
[4]¿Esto necesario? No, es necesario ir en algún orden. No necesariamente este, sin embargo este orden es suficiente.

4. Diseña un algoritmo para encontrar la envolvente convexa de un polígono monótono en tiempo lineal. Un polígono P es monótono respecto a una línea ℓ si cualquier línea perpendicular a ℓ intersecta a P en a lo más dos puntos. Puedes suponer que se conoce ℓ.

Solución. Este problema nos da información extra y valiosa. Esto nos permite modificar el algoritmo de `Graham` para encontrar un orden respecto a los puntos de P en tiempo lineal y reducir nuestra complejidad. A continuación, se exhibe el algoritmo requerido

1. Encontremos un punto distinguido p_0 tal que sea el más "chico" respecto X o Y, por simplicidad este algoritmo asume que p_0 es el más "chico" con respecto a Y. Esto lo encontramos en $\mathcal{O}(n)$.

2. Obtener un orden para poder recorrer. Aprovechando la ℓ-monotonía de nuestro polígono, proyectemos cada punto en ℓ por medio de una transformación $T : (v \in P) \to (v' \in \ell)$. Por tanto, cada proyección nos toma $\mathcal{O}(1)$, como debemos proyectar los n puntos en P entonces gastamos $\mathcal{O}(n)$.

   Ahora, recorramos ℓ (que ya contiene las proyecciones hechas a los puntos en P) por medio de las proyecciones y finalmente obtenemos un orden. Esto nos toma nuevamente $\mathcal{O}(n)$.

   Obs. ¿Qué pasa si existen $u, v \in P$ tal que $T(u) = T(v)$? La respuesta es un poco sencilla, pues basta con encontrar las proyecciones superior e inferior y guardar dos ordenes en cuanto a las proyecciones[5], lo cuál es trivial de obtener al crear una paralela a ℓ, digamos ℓ' tal que P quede entre la manga formada por ℓ y ℓ'. Así, realizamos proyecciones en cada recta (respecto a la cota de inferior y superior respecto de ℓ), recorremos, y obtenemos un orden que podemos fusionar en $\mathcal{O}(1)$. Concluímos que este paso tiene complejidad en $2 \cdot \mathcal{O}(n)$.

3. Por último recorremos el orden en sentido contrario de las manecillas del reloj y recorremos hacia la izquierda, siempre, verificando las direcciones de vuelta. Pues no queremos vueltas a la derecha. Como cada punto conoce a su vecino (por el orden), esto lo podemos hacer en $\mathcal{O}(1)$. Como debemos hacer esto para cada punto, nos toma una complejidad contenida en $\mathcal{O}(n)$. En cada iteración formamos una arista y la guardamos, eventualmente llegaremos a p_0 que es desde dónde inicia nuestro algoritmo.

Análisis de complejidad. Este algoritmo tiene una complejidad en

$$\therefore \mathcal{O}(n) + 2 \cdot \mathcal{O}(n) + \mathcal{O}(n) = 3 \cdot \mathcal{O}(n) \in \mathcal{O}(n).$$

[5] Esto lo decidimos con la coordenada no involcrada en cada de (X, Y, ℓ)-monotonía y para π-monotonía tomamos como parámetro su radio.

5. Sea S un conjunto de n círculos unitarios en el plano que posiblemente se intersectan. Se desea calcular el cierre convexo de S.

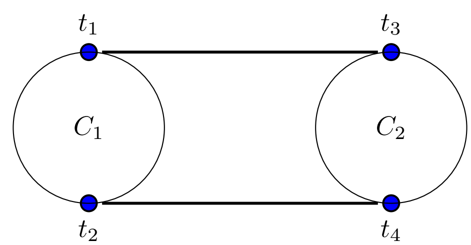
a) Demuestra que el contorno del cierre convexo de S consiste de segmentos de recta y pedacitos de círculos en S.

b) Demuestra que cada círculo puede estar a lo más una vez en el contorno del cierre convexo.

c) Sea S' el conjunto de puntos formado por los centros de los círculos en S. Muestra que un círculo en S aparece en el contorno del cierre convexo sí y sólo sí el centro del círculo pertenece al cierre convexo de S'.

d) Diseña un algoritmo de tiempo $\mathcal{O}(n \log n)$ para calcular el cierre convexo de S.

Dem. (*a*): Procedemos por inducción.
Observemos que cuando tenemos un solo círculo se cumple por vacuidad. Con 2 círculos,



C_1 y C_2, obtenemos tangentes a cada circunferencia, por cada circuferencia en 2 puntos distintos. A partir de los puntos de tangencia con C_1 y hasta la frontera externa tenemos un "pedacito" de circulo (caso análogo con C_2 y t_2, t_4). Las tangentes t_1t_3 y t_2t_4 son las rectas y por tanto terminamos.

H.I. Supongamos que el cierre convexo esta formado por "pedacitos" de círculo y rectas, para k círculos.

¿Qué pasa si introducimos un círculo más? Los posibles casos son:

1. El nuevo círculo queda contenido en el cierre convexo. En este caso y por H.I tenemos que el cierre convexo **No** se ve modificado y terminamos.

2. El nuevo círculo queda como frontera. Por H.I, el cierre ya se forma por "pedacitos" de círculos y segmentos de rectas, además el nuevo círculo anexa
   - una parte de su contorno y dos segmentos tangentes del anterior cierre al nuevo círculo y por tanto terminamos.
   - O es parte de la frontera en un solo punto y por **H.I.** terminamos. □

Dem. (*b*): Procedamos por reducción al absurdo.
Supongamos que un círculo puede estar al menos 2 veces en el contorno del cierre convexo. Entonces, existe un círculo C que tiene al menos **2 arcos** sobre el contorno del cierre convexo. Si esto fuese cierto, tendríamos que hay más de 2 tangentes sobre C tal que **No** se intersectan, como cada tangente a C debe ser parte del contorno, entonces puede pasar que:

1. Hay un círculo que **No** pertenece al cierre, pues es dejado fuera por una tangente (digamos que las dos más juntas hacen el cierre). Hasta aquí hemos llegado a una contradicción por suponer que hay un círculo con al menos dos arcos en el cierre convexo.

2. Los círculos son parte del cierre, y por tanto perdemos convexidad. Nuevamente hemos llegado a contradecir lo supuesto.

¿Por qué perdemos convexidad?
Como hemos dicho, nuestro círculo esta al menos 2 veces en el contorno, esto implica que hay al menos 3 tangentes al círculo que son parte del contorno y que **No** son contiguas (si lo fuera serían una sola recta en vez de 2). Como 2 segmentos **No** son contiguos, entonces, NO existe el caso donde hayan dos rectas contiguas tales que formen un ángulo de 180 y por tanto una recta (o segmento prolongado) **siempre** puede cortar los 3 segmentos tangentes a C!, pues si esto NO sucediera, implicaría que una de las (segmentos) rectas esta contenida por otras 2 y esto es falso, pues supusimos que las 3 son parte del contorno.

$\therefore$ Un círculo esta a lo más una vez en el contorno de cierre convexo.

Dem. (c): Análicemos dos posibles casos:

$\Rightarrow$) Procedemos por reducción al absurdo.

Nota: $C(S)$ el cierre de S y $C(S')$ el cierre de S' Supongamos que en $C(S)$ se encuentra C_i como parte de su contorno y O_i (centro de C_i). No esta $C(S')$, esto implica que:

1. O_i queda fuera de $C(S')$, esto **No** se admite, pues $C(S')$ es el cierre convexo.
2. O_i **No** es parte del contorno de $C(S')$ y queda interno, por tanto existe O_j que es más "externo" que O_i, esto implica que C_j es más externo que C_i (recordando que los círculos son unitarios) pero C_i, es parte del contorno.
   He aquí una contradicción de suponer que $O_i \notin C(S')$.

$\Leftarrow$) Procedamos por reducción al absurdo.

Sea $C(S')$ el cierre convexo de S' y $C(S')$ el cierre convexo de S. Si $O_i \in C(S') \Rightarrow C_i \notin C(S)$ para algún $i \in [1, ..., n]$ y O_i corresponde al centro de C_i.

Como C_i **No** es parte de $C(S)$, entonces existe C_j más "externo" que C_i y por tanto O_j es más "externo" que O_i, pues son círculos unitarios, esto contradice el hecho de que $O_i \in C(S')$ refiriendonos a que forma parte de su contorno, pues O_j queda fuera de $C(S')$ y O_j es elemento de S'. Lo anterior contradice el hecho de que O_i era parte de la frontera de $C(S')$.

De aquí concluimos lo que se quería mostrar. □

Solución (d): Este algoritmo estará basado fuertemente en el algoritmo de Graham. A continuación se exhibe el algoritmo requerido:

1. Aplicar Graham en el conjunto de centros de cada círculos S'. Esto nos toma $O(n \log_2 n)$
2. Obtenemos las paralelas de $C(S')$ en exactamente una unidad, por el inciso c) sabemos que las círcunferencias de estos puntos forman el cierre convexo $C(S)$. Esto nos toma $O(n)$.
3. Por cada recta paralela, digamos $\ell_i's$ tomemos sus extremos en un sentido $\rightarrow$ y recorremos desde ese punto por los pedacitos de círcunferencia a donde son tangentes (por a,b,c) hasta encontrarse con la otra recta, así unimos ese pequeño arco. Esto lo hacemos en $\mathcal{O}(n)$.

El resultado de estos pasos es el cierre convexo de S.

Análisis de complejidad. La complejidad esta contenida en

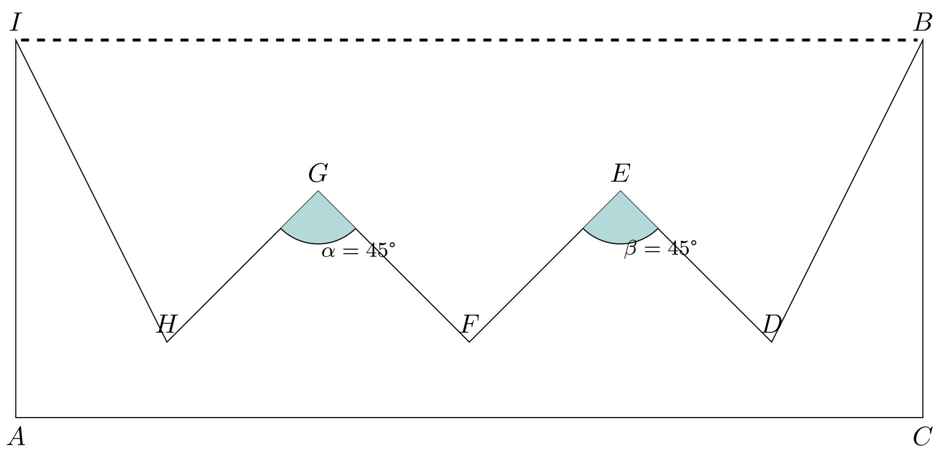
$$\mathcal{O}(n \log n) + \mathcal{O}(n) + \mathcal{O}(n) = \mathcal{O}(n \log n).$$

5. Prueba o da un contra ejemplo:

1. Los vértices convexos de un polígono simple pertenecen al cierre convexo.
2. La intersección de dos polígonos convexos es convexo.
3. La unión de dos polígonos convexos es convexo.
4. La intersección de un polígono estrellado y un convexo es convexo.

Solución:

1. Los vértices convexos de un polígono simple pertenecen al cierre convexo



   α, β son vértices convexos, pues $\alpha, \beta < 180°$ y **NO** forman parte del cierre convexo

2. Procedemos por contradicción.

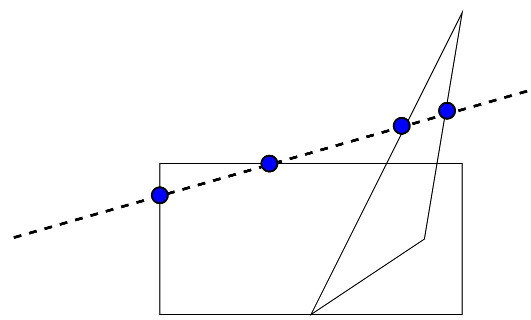
   Supongamos que la intersección de dos polígonos A_i y A_j convexos **No** es convexo. Entonces $A_i \cap A_j$ tienen al menos un ángulo $\beta > 180°$, β debe ser generado por un punto de intersección de $A_i \cap A_j$, pues si no, aplicaría que β viene de un vértice en A_i o A_j, pero estos son convexos.

   Entonces ¿Cómo son las intersecciones $A_i \cap A_j$? (Además son solo 2, por que son convexas). Implica que P_i tal que $\measuredangle P_i > 180°$ tiene como vecinos a vértices de distintos polígonos digamos $P_{Ai} \in A_i$ y $P_{Aj} \in A_j$ entonces el vecino de P_i que **No** es P_{Ai} y esta sobre A_i es un extremo de segmento resultado de prolongar $P_{Ai}P_i$ (pues ya vimos que el ángulo P_i era forrado por intersección) que tiene exactamente 180°, esto indica que P_iP_{Aj} No pertenece a la intersección!.

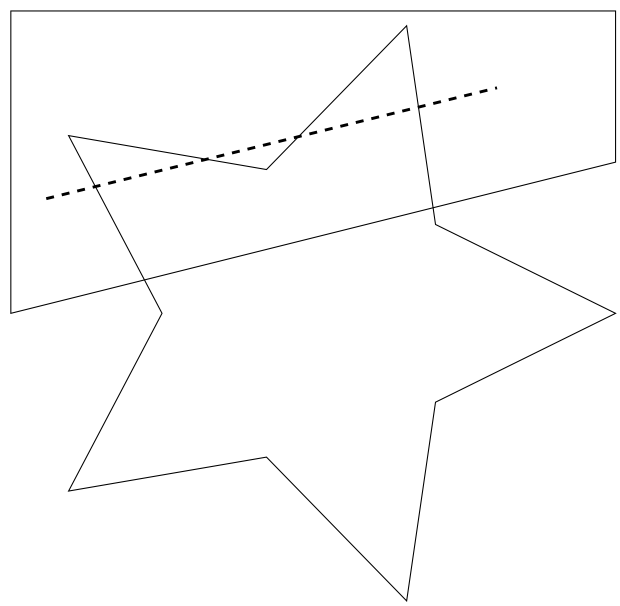
   He aquí una contradicción de suponer que la intersección de polígonos convexos **No** es convexa. Luego concluimos lo que queríamos mostrar.$\square$

3. La unión de dos polígonos convexos es convexo.

   Como podemos ver, las siguientes figuras obedecen a un rectángulo y un triángulo, ambos convexos. Sin embargo, su unión no es convexa, pues la línea punteada corta esta unión en 4 puntos. Recordemos que un polígono convexo es cortado a lo más en 2 puntos, es por esto que el enunciado es falso.



4. La intersección de un polígono estrellado y un convexo es convexo.



Como podemos notar, la recta punteada corta al polígono resultado de la intersección en 4 puntos, por esta razón el polígno resultado de la intersección antes mencionada **No** es convexo.

7. Dado un conjunto de triángulos (definiendo sus 3 puntos) describe un algoritmo de barrido de línea que encuentre los triángulos contenidos en otros.

Solución. Realicemos barrido de línea horizontal.

1. Ordenemos nuestro conjunto de puntos $\mathcal{O}(n \log n)$.
2. Empezamos el barrido desde el punto más a la izquierda toma $\mathcal{O}(n)$ encontrar el punto más a la izquierda.
3. Iniciamos barriendo tal como lo realizamos en el barrido de línea para encontrar intersecciones, con la modificación, que **No** sacamos segmentos de nuestro árbol AVL, en cambio sacamos triángulos.
4. Por cada vértice en donde inicie un triángulo verificamos el estado de la línea, si hay para ver si ese punto esta contenido en otro triángulo (el que este en el estado), esto es fácil de verificar, pues los triángulos ya están definidos. Si el vértice esta contenido en uno o más triángulos, verificamos el resto de vértices.

Si No esta contenido en algún triángulo, el triángulo se anexa al estado de la línea y nos vemos al siguiente vértice.

Observación En los 2 casos metemos el triángulo al estado de la línea.

Esto nos puede llevar $\mathcal{O}(\frac{n}{3}) \in \mathcal{O}(n)$. Como esto se realiza para al menos $\frac{n}{3}$ vértices (los que inician los triángulos), entonces tenemos una complejidad total de $\mathcal{O}(n^2)$.

Análisis de complejidad. La complejidad esta contenida en

$$\mathcal{O}(n \log n) + \mathcal{O}(n) + \mathcal{O}(n^2) = \mathcal{O}(n^2).$$

8. Suponga que se tiene una lista de aristas doblemente ligada (DCEL) de una subdivisión plana. Diseña un algoritmo que encuentre todas las caras que contienen un vértice en la cara exterior.

Solución. A continuación se exhibe el algoritmo requerido:

1. Ordenamos la lista de aristas (DCEL), esto nos toma $\mathcal{O}(n \log n)$.

2. Encontremos el punto más "chico" en nuestro polígono. Esto lo realizamos en $\mathcal{O}(1)$, pues hemos ordenado nuestras aristas.

3. Recorramos siempre hacia la izquierda excepto cuando:

   (a) El vértice al que llegamos no pertenece a una cara externa. En este caso nos regresamos y ahora tomamos el camino más a la derecha.

   (b) El vértice de llegada es el vértice de partida, entonces hemos llegado y terminamos.

   Durante el recorrido vamos acumulando los vértices que sean externos en una lista, como el recorrido esta dado por las aristas DCEL esto nos toma a lo más tiempo $\mathcal{O}(n)$. Pues cuando lleguemos a un vértice interno de inmediato nos regresamos.

   ¿Cómo sabemos que el vértice pertenece o no a la cara externa? La arista nos lo indica.

Análisis de complejidad. La complejidad de este algoritmo esta contenida en

$$\mathcal{O}(n \log n) + \mathcal{O}(n) + \mathcal{O}(1) = \mathcal{O}(n \log n).$$