

Estructura de datos y teoría de algoritmos

Tarea 4

Fabián Romero

November 19, 2013

1 Reporte de lectura

- a En el artículo de Griggs y Reid an Australasian Journal of Combinatorics 20 (1999), pp 19-24. Se describe la condición para que una secuencia sea de puntajes sea un torneo, en términos de coeficientes binomiales. Demuestra en detalle que si una secuencia es de puntajes de un torneo, entonces satisface la condición.

2 Divide y vencerás

Problema 1 El cierre convexo de un conjunto de S de n puntos en el plano, es una secuencia h_1, h_2, \dots, h_z de puntos de S en posición convexa, que forman los vértices del polígono convexo más pequeño que contiene en su interior o frontera a todos los puntos de S . Diseña un algoritmo divide y vencerás de tiempo $O(n \log n)$ que tome como entrada un conjunto S de n puntos representados como pares de coordenadas (x, y) y calcule el cierre convexo de S . Muestra una ejecución de tu algoritmo con $n = 16$ puntos. Puedes suponer que tu conjunto de entrada no tiene a tres puntos que son colineales.

```
#!/usr/bin/env python

def x(point): return point[0]
def y(point): return point[1]

def derecha((p1,p2,p3)):
    return (x(p2)*y(p3)+x(p1)*y(p2)+
            x(p3)*y(p1)-x(p2)*y(p1)-
            x(p3)*y(p2)-x(p1)*y(p3))<1

def puntos_a_la_derecha(puntos):
    c = [puntos[0], puntos[1]]
    for p in puntos[2:]:
        c.append(p)
        while len(c) > 2 and not derecha(c[-3:]):
            del c[-2]
    return c

def cierre_convexo(puntos):
    puntos.sort()
    arriba=puntos_a_la_derecha(puntos)
    abajo=puntos_a_la_derecha(puntos[::-1])[1:-1]
    return arriba + abajo
```

Demostración Queremos demostrar que:

El cierre convexo de un conjunto de n puntos en el plano se puede calcular en el tiempo $O(n\log(n))$.

Vamos a demostrar, la parte superior del cierre , la parte inferior se demuestra análogamente.

Observe que la parte superior de un cierre convexo que empieza por el punto con menor coordenada x será una trayectoria formada por segmentos unidos dos a dos que solo tiene giros a la derecha que llamaremos cadena y que todos los puntos que no pertenezcan a la cadena estarán por debajo de ella.

La prueba es por inducción sobre el número de puntos.

En la función *puntos_a_la_derecha* Antes de que comience el *for*, la lista contiene los puntos p_1 y p_2 que para el caso de longitud 2 son claramente su propio cierre convexo. (caso base)

Supongamos ahora por hipotesis de inducción que en la lista $arriba_{i-1}$ se calcula correctamente los vértices superiores del cierre convexo de $\{p_1, \dots, p_{i-1}\}$ y consideramos la adición de p_i para demostrar el caso inductivo.

Después de la ejecución del *while* y por la hipótesis de inducción, sabemos que los puntos en la lista $arriba_{i-1}$ forman el cierre superior y son una cadena que sólo tiene giros a la derecha. Por otra parte, la cadena se inicia en el punto de menor coordenada x de p_1, \dots, p_i y termina en el de mayor coordenada x , es decir, P_i . Demostremos que todos los puntos de p_1, \dots, p_i que no están en la lista $arriba_i$ están por debajo de la cadena. Por inducción sabemos que no hay punto por encima de la cadena que teníamos antes que añadiéramos P_i , observe que el algoritmo itera secuencialmente sobre p_1, \dots, p_i , por lo que la única forma de que un punto no esté por debajo de $arriba_i$ es en el tramo vertical entre p_{i-1} y p_i . Pero esto no es posible, ya que tal punto estaría en entre p_{i-1} y p_i y todos esos están bajo la cerradura convexa de la hipótesis de inducción.

Para probar la cota $O(n \log(n))$. Primero ordenamos los puntos con respecto a x , que sabemos se puede hacer en $O(n \log(n))$ Consideremos ahora el cálculo de la parte superior del cierre. El ciclo *for* se ejecuta linealmente sobre los puntos. La pregunta que queda es la frecuencia con que se ejecuta el bucle *while* interior. Para cada ejecución del *for* el bucle *while* se ejecuta al menos una vez. Para cualquier ejecución extra un punto se elimina del cierre calculado, dado que cada punto se elimina cuando más una vez durante la construcción de la parte superior del cierre, el número total de ejecuciones extra es cuando mas n por lo que el tiempo total es $O(n \log(n)) + O(n) = O(n \log(n))$. que nos indujo el ordenar los puntos, una vez ordenados, el tiempo es lineal.

■

Problema 2 En la primaria aprendiste un algoritmo de multiplicación de secuencias de n dígitos de complejidad $\theta(n^2)$. Ahora diseña un algoritmo que tome como entrada dos cadenas S_x, S_y denbitscadauna, y calcule $\text{tiempo}(n^2)$

la cadena de $2n$ bits que representa al producto de los números x y y , donde x es la concatenación de los bits de S_x y y es la concatenación de S_y . Muestra una ejecución de tu algoritmo con dos números de 16 bits.

3 Programación Dinámica