Licenciatura en ciencia de la computación

Ciclo minimo y Ciclo de Hamilton

Matemática Computacional

Profesor:
Nicolas Thériault

Autor:
Sergio Salinas
Danilo Abellá

Contents

1 Introducción

Este informe tratara sobre el analisis de tres algoritmos a la vez. El algoritmo de busca de ciclo minimo, que buscara cuál es el ciclo que contiene menos vertices en un grafo. El algoritmo de ver si un grafo un convexo y por último la busqueda del camino hamiltoniano, este trata de buscar un camino dentro del grafo que pase por todos los vertices y se devuelva por el vertice de inicio.

2 Explicación algoritmo

El algoritmo utiliza una matriz de nxn para representar el grafo, donde "n" es la cantidad de vértices y "p" de aristas.

Para saber si existe un ciclo Hamilton en el grafo primero se recorre desde un vértice "0" y se va avanzando por todos los vértices que compartan aristas (sin repetir vértices), si al terminar el recorrido se pudo recorrer todos los vértices del grafo (sin repetirse) y regresar al primer vértice: se afirma que es ciclo hamilton y se muestra en pantalla dicho ciclo.

En caso de llegar a un vértice que no sea el inicial y no tenga mas aristas que lleven a nuevos vértices que recorrer: retrocede al vértice anterior y busca otro camino posible, si nuevamente no encuentra, vuelve a retroceder y buscar otro vértice distinto nuevamente, y así susesivamente.

Si al termina el recorrido y no se encuentra ningún ciclo hamilton, se muestra en pantalla cual fue el último movimiento realizado y hasta que vértice llegó, mostrando al final una "X" donde NO hay mas vértices nuevos que avanzar.

Todo esto representando el grafo en una matriz donde las filas y columnas son los vértices, y sus posiciones representan si están unidas por una arista o no, en la cual "0" afirma no tener aristas en común y "1" si.

Para el ciclo mínimo se recorre el grafo casi de igual manera, nada mas que si puede repetir un vértice y cerrar ahí un ciclo, éste se guarda y se compara en caso de que aparezcan otros ciclos, el ciclo de menor cantidad de vértices (mínimo 3 vértices) se muestra en pantalla.

3 Formulación experimentos

Para analizar los algoritmos se probo con con 4 valores distintos de n, n= 100, 500, 1000, 10000, en los que se variaba los valores de p.

El rango de los valores de p iba desde 0 hasta 1, con intervalos de 0.01, por lo que cada n se probaba con 100 valores de p distintos, la tabla de resultados se puede ver en la capeta $informe/plots/$, donde la primera columna es el valor p y la segunda el tiempo

Como los tres algoritmos dependendian del otro para funcionar se decicio hacer las tablas y los graficos del ficionamientos de los tres algoritmos juntos.

4 Información de Hardware y Software

4.1 Notebook - Danilo Abellá

4.1.1 Software

- SO: Xubuntu 16.04.1 LTS
- GMP Library
- Mousepad 0.4.0

4.1.2 Hardware

- AMD Turion(tm) X2 Dual-Core Mobile RM-72 2.10GHz
- Memoria (RAM): 4,00 GB(3,75 GB utilizable)
- Adaptador de pantalla: ATI Raedon HD 3200 Graphics

4.2 Notebook - Sergio Salinas

4.2.1 Software

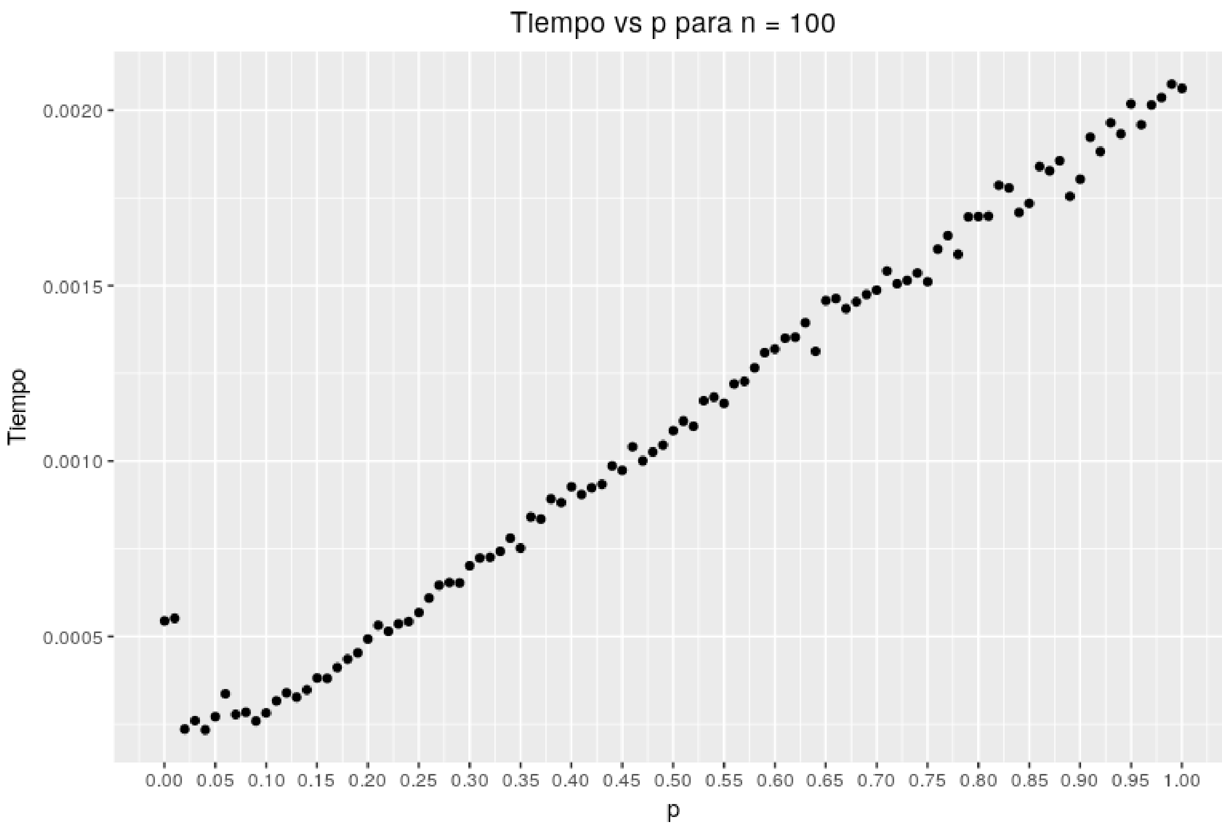
- SO: ubuntu Gnome 16.04 LTS
- Compilador: gcc version 5.4.0 20160609
- Editor de text: Atom

4.2.2 Hardware

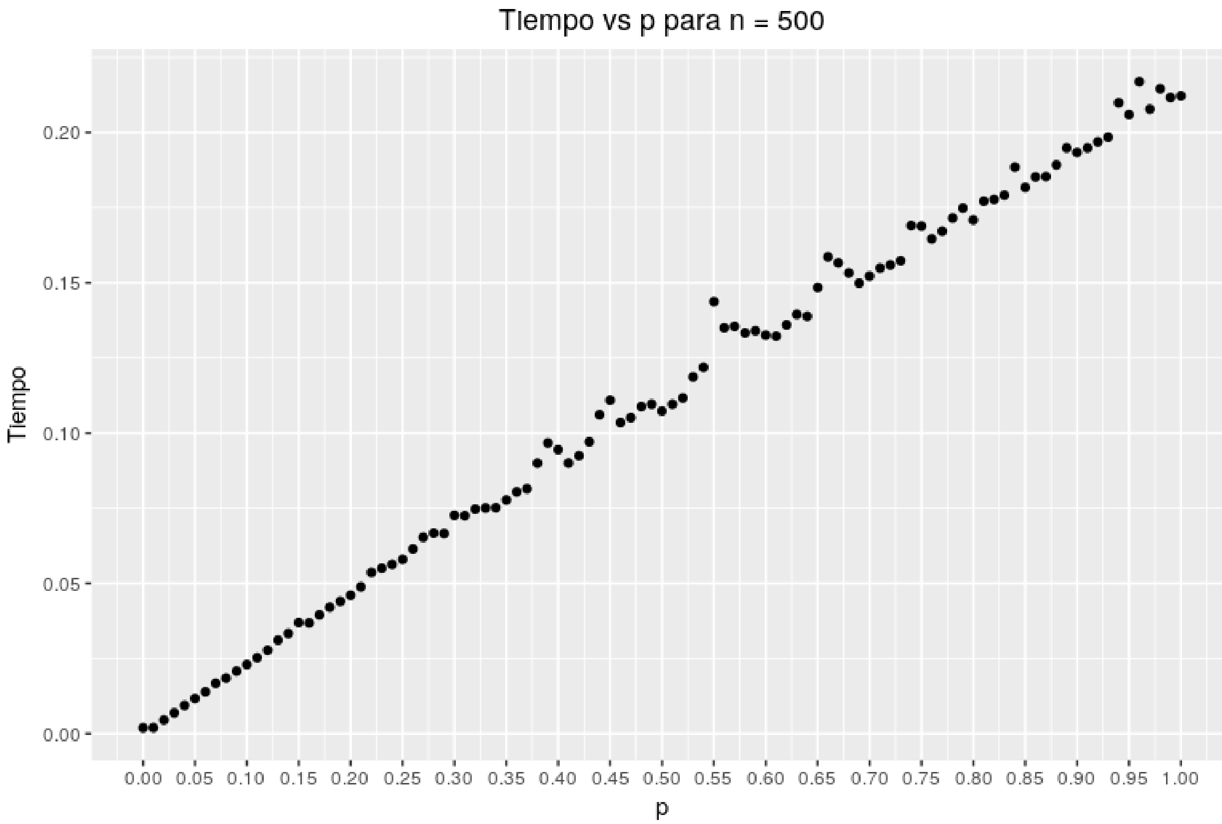
- Procesador: Intel Core i7-6500U CPU 2.50GHz x 4
- Video: Intel HD Graphics 520 (Skylake GT2)

5 Curvas de desempeño de resultados

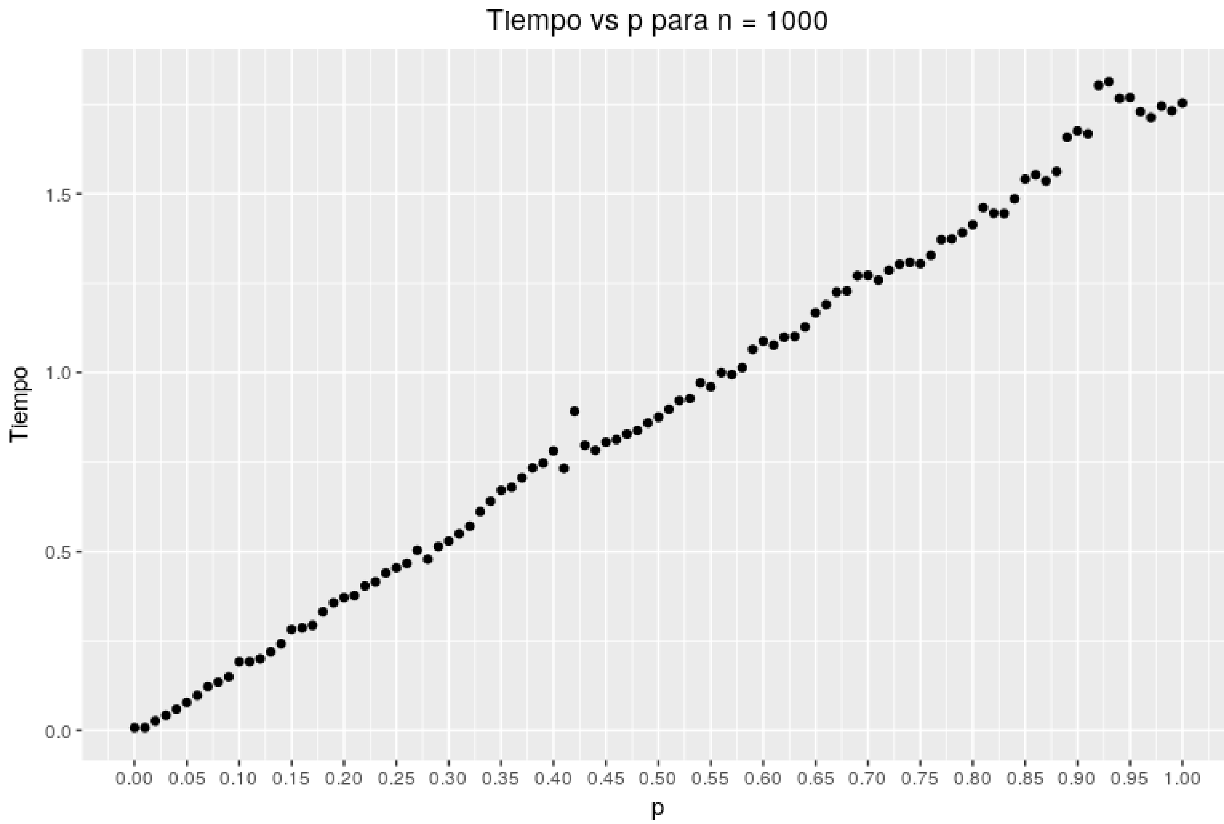
5.1 n = 100



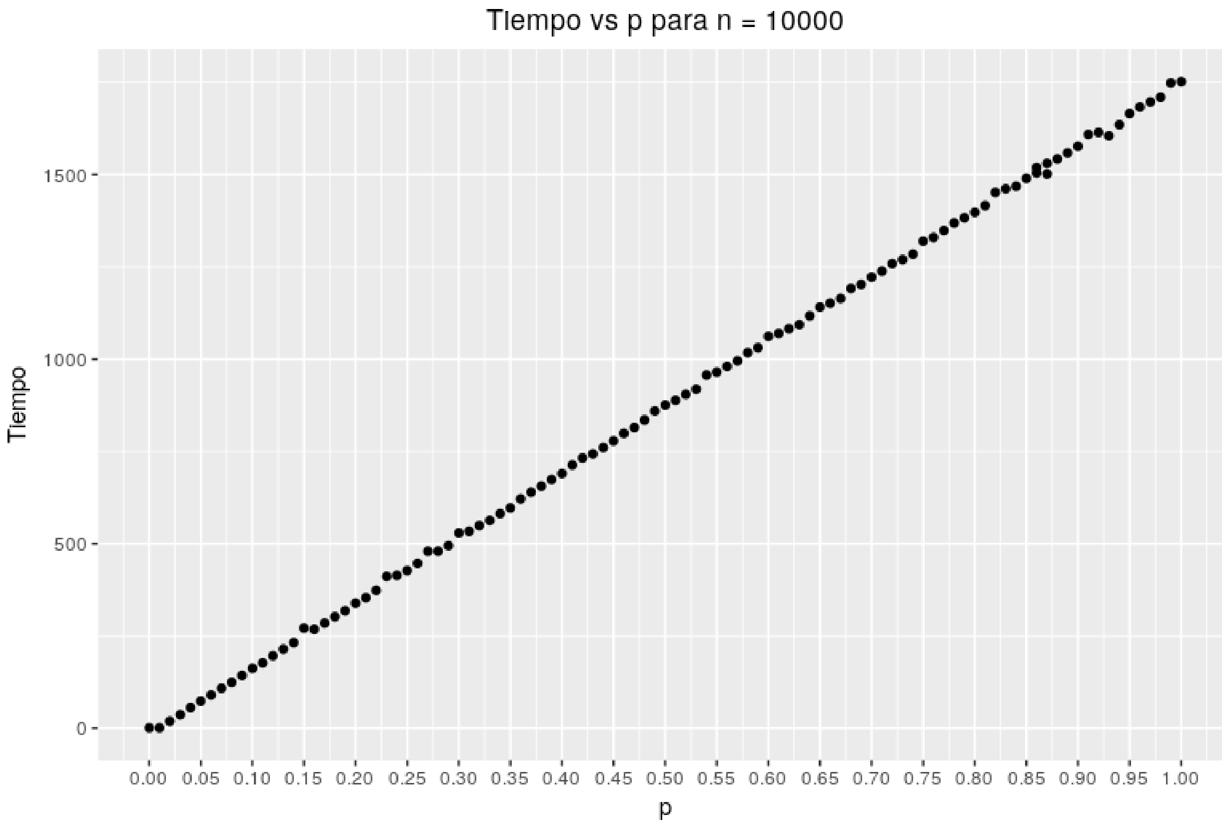
5.2 n = 500


Tiempo vs p para n = 500
Tiempo
0.00
0.05
0.10
0.15
0.20
p
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00

5.3 n = 1000


Tiempo vs p para n = 1000
Tiempo
p
0.0
0.5
1.0
1.5
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00

5.4 n = 10000


Tiempo vs p para n = 10000
Tiempo
1500
1000
500
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00
p

6 Conclusiones

Se puede apreciar que ha mayor valor p mayor coste computacional tendrá el algoritmo, pero en contraste los tres primeros gráficos con n = 100, 500 y 1000 se mantuvieron similares en el tiempo, aun así con el último gráfico el tiempo crecio, por lo que a mayor valor de n también es mayor el tiempo de ejecución.

Por lo tanto se puede concluir que a mayor cantidad de vértices y aristas tenga un grafo, mayor será la cantidad de tiempo que se demorara el algoritmo en encontrar un ciclo minimo y un ciclo hamiltoniano.