Introducción a Algoritmos en Grafos

Miguel Raggi
mraggi@gmail.com

Algoritmos en Grafos
Escuela Nacional de Estudios Superiores
UNAM

29 de enero de 2018

Introducción

- Bienvenidos a algoritmos en grafos con Raggi-Sensei!

Introducción

- Bienvenidos a algoritmos en grafos con Raggi-Sensei!
- Vamos a aprender muuucho.

Programa

1. Introducción
2. Estructuras de datos para grafos
3. Generación de grafos aleatorios
4. Árbol generador de peso mínimo
5. Caminos en grafos
6. Programación Lineal
7. Apareamientos y Flujos
8. Algoritmos en digrafos
9. Redes (centralidad, epidemias, percolación, etc.)
10. NP-completos

¿Qué aprenderemos (además de algoritmos)?

Mis objetivos, más que enseñarles una colección de algoritmos, son:

¿Qué aprenderemos (además de algoritmos)?

Mis objetivos, más que enseñarles una colección de algoritmos, son:

- Que aprendan a hacer programas grandes y con muchas componentes que se conectan.

¿Qué aprenderemos (además de algoritmos)?

Mis objetivos, más que enseñarles una colección de algoritmos, son:

- Que aprendan a hacer programas grandes y con muchas componentes que se conectan.
- Todo el curso haremos un solo programa.

¿Qué aprenderemos (además de algoritmos)?

Mis objetivos, más que enseñarles una colección de algoritmos, son:

- Que aprendan a hacer programas grandes y con muchas componentes que se conectan.
- Todo el curso haremos un solo programa.
- Que aprendan bien test-driven-development.

¿Qué aprenderemos (además de algoritmos)?

Mis objetivos, más que enseñarles una colección de algoritmos, son:

- Que aprendan a hacer programas grandes y con muchas componentes que se conectan.
- Todo el curso haremos un solo programa.
- Que aprendan bien test-driven-development.
- Que manejen grafos, algoritmos y estructuras de datos.

¿Qué aprenderemos (además de algoritmos)?

Mis objetivos, más que enseñarles una colección de algoritmos, son:

- Que aprendan a hacer programas grandes y con muchas componentes que se conectan.
- Todo el curso haremos un solo programa.
- Que aprendan bien test-driven-development.
- Que manejen grafos, algoritmos y estructuras de datos.
- No alcanzaremos a ver todo lo anterior.

¿Qué aprenderemos (además de algoritmos)?

Mis objetivos, más que enseñarles una colección de algoritmos, son:

- Que aprendan a hacer programas grandes y con muchas componentes que se conectan.
- Todo el curso haremos un solo programa.
- Que aprendan bien test-driven-development.
- Que manejen grafos, algoritmos y estructuras de datos.
- No alcanzaremos a ver todo lo anterior. No pasa nada!

¿Qué usaremos?

Muestra!

Calificación

- 60 % de su calificación será su proyecto del semestre, que estarán construyendo individualmente durante todo el semestre.

Calificación

- 60 % de su calificación será su proyecto del semestre, que estarán construyendo individualmente durante todo el semestre.
  - 30 % si está correcto

Calificación

- 60 % de su calificación será su proyecto del semestre, que estarán construyendo individualmente durante todo el semestre.
  - 30 % si está correcto
  - 30 % si está bonito programado, legible, tiene suficientes casos de prueba, etc.

Calificación

- 60 % de su calificación será su proyecto del semestre, que estarán construyendo individualmente durante todo el semestre.
  - 30 % si está correcto
  - 30 % si está bonito programado, legible, tiene suficientes casos de prueba, etc.
- 30 % será su calificación de hackerrank, donde les dejaré problemas extra.

Calificación

- 60 % de su calificación será su proyecto del semestre, que estarán construyendo individualmente durante todo el semestre.
  - 30 % si está correcto
  - 30 % si está bonito programado, legible, tiene suficientes casos de prueba, etc.
- 30 % será su calificación de hackerrank, donde les dejaré problemas extra.
- 20 % será su exposición.

Calificación

- 60 % de su calificación será su proyecto del semestre, que estarán construyendo individualmente durante todo el semestre.
  - 30 % si está correcto
  - 30 % si está bonito programado, legible, tiene suficientes casos de prueba, etc.
- 30 % será su calificación de hackerrank, donde les dejaré problemas extra.
- 20 % será su exposición.
- Warning: Todo lo que veremos ya está programado por alguien. La idea es volverlo a hacer, sin hacer trampa.

Calificación

- 60 % de su calificación será su proyecto del semestre, que estarán construyendo individualmente durante todo el semestre.
  - 30 % si está correcto
  - 30 % si está bonito programado, legible, tiene suficientes casos de prueba, etc.
- 30 % será su calificación de hackerrank, donde les dejaré problemas extra.
- 20 % será su exposición.
- Warning: Todo lo que veremos ya está programado por alguien. La idea es volverlo a hacer, sin hacer trampa.
- El lenguaje que usaremos será C++. De vez en cuando pondré ejemplitos en sage.

Calificación

- 60 % de su calificación será su proyecto del semestre, que estarán construyendo individualmente durante todo el semestre.
  - 30 % si está correcto
  - 30 % si está bonito programado, legible, tiene suficientes casos de prueba, etc.
- 30 % será su calificación de hackerrank, donde les dejaré problemas extra.
- 20 % será su exposición.
- Warning: Todo lo que veremos ya está programado por alguien. La idea es volverlo a hacer, sin hacer trampa.
- El lenguaje que usaremos será C++. De vez en cuando pondré ejemplitos en sage.

Pero... ¡yo sólo se python!

Pero... ¡yo sólo se python!

- Pues aprende algo nuevo, no te hará daño.

Pero... ¡yo sólo se python!

- Pues aprende algo nuevo, no te hará daño.
- En hackerrank puedes usar el lenguaje que quieras.

Pero... ¡yo sólo se python!

- Pues aprende algo nuevo, no te hará daño.
- En hackerrank puedes usar el lenguaje que quieras.
- En el proyecto también, pero tendrá que tener toda la misma funcionalidad que el programa que les pasaré en C++, y tú mismo te tendrás que encargar de que funcione.

Pero... ¡yo sólo se python!

- Pues aprende algo nuevo, no te hará daño.
- En hackerrank puedes usar el lenguaje que quieras.
- En el proyecto también, pero tendrá que tener toda la misma funcionalidad que el programa que les pasaré en C++, y tú mismo te tendrás que encargar de que funcione.
- Tarea de aquí a que salgas de la licenciatura (y después): Tutoriales de C++. Todos los días.

Pero... ¡yo sólo se python!

- Pues aprende algo nuevo, no te hará daño.
- En hackerrank puedes usar el lenguaje que quieras.
- En el proyecto también, pero tendrá que tener toda la misma funcionalidad que el programa que les pasaré en C++, y tú mismo te tendrás que encargar de que funcione.
- Tarea de aquí a que salgas de la licenciatura (y después): Tutoriales de C++. Todos los días.
- Cuando ya sientas que sabes mucho C++ (o incluso al mismo tiempo), puedes empezar con otro. Por ejemplo, D o Kotlin o Julia o Go o Haskell.

Pero... ¡yo sólo se python!

- Pues aprende algo nuevo, no te hará daño.
- En hackerrank puedes usar el lenguaje que quieras.
- En el proyecto también, pero tendrá que tener toda la misma funcionalidad que el programa que les pasaré en C++, y tú mismo te tendrás que encargar de que funcione.
- Tarea de aquí a que salgas de la licenciatura (y después): Tutoriales de C++. Todos los días.
- Cuando ya sientas que sabes mucho C++ (o incluso al mismo tiempo), puedes empezar con otro. Por ejemplo, D o Kotlin o Julia o Go o Haskell. Ya basta de sólo saber uno.

¿Qué tecnologías usaremos?

- C++, git, cmake, clang-tidy, clang-format, travis-ci

¿Qué tecnologías usaremos?

- C++, git, cmake, clang-tidy, clang-format, travis-ci
- SFML: Para dibujar

¿Qué tecnologías usaremos?

- C++, git, cmake, clang-tidy, clang-format, travis-ci
- SFML: Para dibujar
- Google Testing Framework: Para hacer test-driven-development.

¿Qué tecnologías usaremos?

- C++, git, cmake, clang-tidy, clang-format, travis-ci
- SFML: Para dibujar
- Google Testing Framework: Para hacer test-driven-development.
- Hackerrank: Para tareas y eso

¿Qué tecnologías usaremos?

- C++, git, cmake, clang-tidy, clang-format, travis-ci
- SFML: Para dibujar
- Google Testing Framework: Para hacer test-driven-development.
- Hackerrank: Para tareas y eso
- Sagemath: Para hacer prototipados, muestras, y pruebas.

Ya que sepan sintaxis básica

- Les pasaré individualmente un libro bueno y algunas ligas con tutoriales, aunque lo mejor es que ustedes mismos busquen y escojan uno que se adapte a si mismos.

Ya que sepan sintaxis básica

- Les pasaré individualmente un libro bueno y algunas ligas con tutoriales, aunque lo mejor es que ustedes mismos busquen y escojan uno que se adapte a si mismos.
- Pero de todos modos, yo estoy aquí para contestarles cualquier pregunta.

Ya que sepan sintaxis básica

- Les pasaré individualmente un libro bueno y algunas ligas con tutoriales, aunque lo mejor es que ustedes mismos busquen y escojan uno que se adapte a si mismos.
- Pero de todos modos, yo estoy aquí para contestarles cualquier pregunta.
- Además, yo les daré tutorial de kdevelop, cmake, git, sfml y google test, para que puedan correr el programa.

Ya que sepan sintaxis básica

- Les pasaré individualmente un libro bueno y algunas ligas con tutoriales, aunque lo mejor es que ustedes mismos busquen y escojan uno que se adapte a si mismos.
- Pero de todos modos, yo estoy aquí para contestarles cualquier pregunta.
- Además, yo les daré tutorial de kdevelop, cmake, git, sfml y google test, para que puedan correr el programa.

Laptops

Traigan su laptop para la próxima clase, con lo siguiente instalado:

Laptops

Traigan su laptop para la próxima clase, con lo siguiente instalado:

- Kdevelop (sí hay para windows y mac) o de perdis Codeblocks más o menos nuevo.
- cmake
- git
- clang y clang-tools
- SFML
- google test

Laptops

Traigan su laptop para la próxima clase, con lo siguiente instalado:

- Kdevelop (sí hay para windows y mac) o de perdis Codeblocks más o menos nuevo.
- cmake
- git
- clang y clang-tools
- SFML
- google test

Ya con eso, aquí les ayudo a instalar el programita que hice, y comenzaremos por empezar a agregarle funcionalidad poco a poco.

Laptops

Traigan su laptop para la próxima clase, con lo siguiente instalado:

- Kdevelop (sí hay para windows y mac) o de perdis Codeblocks más o menos nuevo.
- cmake
- git
- clang y clang-tools
- SFML
- google test

Ya con eso, aquí les ayudo a instalar el programita que hice, y comenzaremos por empezar a agregarle funcionalidad poco a poco. Primero entre todos (yo lo pongo y uds. lo copian) y luego ya individualmente.

Laptops

Traigan su laptop para la próxima clase, con lo siguiente instalado:

- Kdevelop (sí hay para windows y mac) o de perdis Codeblocks más o menos nuevo.
- cmake
- git
- clang y clang-tools
- SFML
- google test

Ya con eso, aquí les ayudo a instalar el programita que hice, y comenzaremos por empezar a agregarle funcionalidad poco a poco. Primero entre todos (yo lo pongo y uds. lo copian) y luego ya individualmente.