Bienvenida

¡Bienvenido al libro online de Matemáticas Discretas para Ciencia de Datos!

Preámbulo

Dos de las habilidades complementarias de mayor utilidad en una carrera de Ciencia de Datos son el uso de teoría de gráficas y el diseño de algoritmos. Sin embargo, usualmente se requiere de mucho tiempo para entender con profundidad los fundamentos matemáticos y de ciencias de la computación detrás de estos temas. En ocasiones, entender todos los detalles requiere de tomar varios cursos a nivel universitario. Si bien esto llevaría a un entendimiento a detalle de la teoría, usualmente no hay el tiempo necesario para aprender todo esto dentro de los ambiciosos programas de estudio en ciencias de datos.

El objetivo de este libro es dar una muy buena introducción a estos temas. El tratamiento no tiene la profundidad de tomar varios cursos completos. Sin embargo, el material sí es suficiente para desarrollar un fuerte nivel de abstracción y para cubrir la mayoría de las necesidades prácticas de un científico de datos.

Se cubre el material desde dos perspectivas. Por un lado, se desarrolla teoría matemática y computacional de manera formal, dando definiciones, ejemplos, proposiciones, teoremas y demostraciones cada que sea pertinente. Por otro lado, se desarrolla el aspecto práctico mediante descripciones de algoritmos, pseudocódigos y numerosas celdas para ejecutar. El lenguaje de programación elegido es Python y se usan varias librerías abiertas.

Al final de cada tema se presenta una sección de Tarea Moral, cuyo objetivo es continuar practicando de manera autodidacta el contenido cubierto. Además, al final de cada capítulo se presenta una lista de ejercicios teóricos y de implementación.

Pre-requisitos

En términos de matemáticas, supondremos que el lector tiene una formación buena correspondiente a los primeros dos semestres de una licenciatura en matemáticas o algo afín. Específicamente, supondremos cierta familiaridad con los temas de inducción, sucesiones, conteo y cálculo. Supondremos también que el lector está acostumbrado a seguir una estructura de definiciones, proposiciones, lemas, teoremas, corolarios. En términos de computación teórica no supondremos pre-requisitos.

La mayor parte del texto puede leerse sólo de manera teórica. Sin embargo, para entender con profundidad los temas y llevarlos a la práctica, recomendamos que el lector estudie y juegue con todo el código en Python que se propone a lo largo del libro. Para ello, supondremos un manejo básico de Python: tipos de datos, asignaciones, comparaciones, ciclos, condicionales, etc.

Temario

> **Nota**
>
> Este es un libro en elaboración. En el siguiente temario puedes encontrar indicados los capítulos con más progreso.

- Parte 1: Fundamentos combinatorios
  - ☑ Buscar un patrón
  - ☑ Principio de las casillas
  - ☑ Principio de doble conteo y coeficientes binomiales
  - ☑ Principio de inducción
  - ☑ Principio de recursión y recursiones lineales
  - ☑ Principio extremo

- [x] Principio extremo

- Parte 2: Teoría de gráficas
  - [x] Gráficas, grados y lema de Euler
  - [x] Caminos, conexidad y distancia
  - [x] Caminos cerrados, circuitos y ciclos
  - [x] Recorrer toda una gráfica
  - [x] Árboles y bosques
  - [x] Gráficas bipartitas
  - [x] Emparejamientos y teorema de Hall
  - [x] Conjuntos independientes y coloraciones
  - [x] Coloración de aristas y teorema de Ramsey
  - [ ] Subgráficas completas y teorema de Turán
- Parte 3: Diseño y análisis de algoritmos
  - [x] Problemas y algoritmos
  - [x] Algoritmos incorrectos
  - [x] Algoritmos correctos
  - [x] Modelo RAM y pensamiento asintótico
  - [x] Notación O grande y similares
  - [x] Propiedades de la notación O grande
  - [x] Ejemplos de análisis de eficiencia
  - [x] El problema de ordenar
  - [x] Ordenar cuadráticamente
  - [x] Estructuras de datos vs tipos de datos abstractos
  - [x] Diccionarios y árboles de búsqueda balanceados
  - [x] Ordenar eficientemente
  - [x] Aplicaciones de ordenar
- Parte 4: Heurísticas de creación de algoritmos
  - [x] Tipos de problemas algorítmicos
  - [ ] Espacio de estados
  - [ ] Exploración exhaustiva
  - [ ] Recortes al espacio de estados
  - [ ] Algoritmos voraces
  - [ ] Divide y conquista
  - [ ] Recursión y teorema maestro
  - [ ] Backtrack en búsquedas combinatorias
  - [ ] Más ejemplos de backtrack
  - [ ] Programación dinámica
  - [ ] Métodos probabilistas
- Parte 5: Algoritmos en teoría de gráficas
  - [x] Implementaciones de gráficas y variantes
  - [x] Uso básico de NetworkX
  - [x] Búsqueda por anchura
  - [x] Aplicaciones de búsqueda por anchura
  - [x] Búsqueda por profundidad

- [ ] Aplicaciones de búsqueda por profundidad
- [x] Árboles de peso mínimo: algoritmos de Prim y Kruskal
- [ ] Caminos de peso mínimo: algoritmos de Dijkstra y Floyd-Warshall
- [ ] Redes, flujos y flujos máximos
- [ ] Algoritmos de Ford-Fulkerson y Edmonds-Karp
- [x] Reducciones algorítmicas y clases P, NP y NP-completo
- [ ] Problemas de gráficas NP-completos

Siguiente
Fundamentos de combinatoria
›

Por Leonardo Ignacio Martínez Sandoval