СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛИМЕНТ ОХРИДСКИ”, ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

КОНСПЕКТ НА ИЗБИРАЕМАТА УЧЕБНА ДИСЦИПЛИНА “ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ”

(зимен семестър на 2016 / 2017 уч. г.)

1. Алгоритми — неформално определение. Изчислителни задачи. Дължина на входа на алгоритъм. Сложност на алгоритъм — сложност по време и сложност по памет; сложност в най-добрия и в най-лошия случай, средна сложност; амортизирана сложност.
2. Асимптотичен анализ на сложността — предимства и недостатъци. Нотации, използвани при асимптотичния анализ: $O, o, \Theta, \Omega, \omega$. Свойства на асимптотичните отношения.
3. Методи за доказване на коректност на алгоритъм — инварианта на цикъл, математическа индукция и др.
4. Сортиране — приложения, важност, примери за използване на сортирането като първи етап от решението на други задачи. Устойчиви и неустойчиви алгоритми за сортиране.
5. Елементарни сортиращи алгоритми — метод на мехурчето, сортиране чрез пряко търсене, сортиране чрез вмъкване. Анализ на елементарните сортиращи алгоритми — коректност, устойчивост, сложност по време и по памет.
6. Двоична пирамида. Двоичните пирамиди като двоични дървета и като масиви. Построяване на двоична пирамида с HeapInsert.
7. Построяване на двоична пирамида чрез процедурата Heapify. Сравнителен анализ на начините за построяване на пирамида. Процедура BuildHeap; анализ на нейната времева сложност: наивна реализация ($\Theta(n \log n)$) и бърза реализация ($\Theta(n)$). Пирамидално сортиране.
8. Приоритетната опашка като абстрактен тип данни. Сравнителен анализ на две възможни реализации на приоритетна опашка — чрез обикновен масив и чрез двоична пирамида.
9. Рекурсивни алгоритми и рекурентни уравнения. Методи за решаване на рекурентни уравнения: математическа индукция, развиване на уравнението, дърво на рекурсията, мастър-теорема, характеристично уравнение.
10. Алгоритмична схема “разделяй и владей” — идея на метода и област на приложение. Сортиране чрез сливане — коректност, устойчивост и сложност по време и памет. Бързо умножение.
11. Бързо сортиране. Стандартна реализация — коректност и устойчивост; максимална и средна сложност по време и памет. Подобрения: намаляване на количеството използвана памет чрез опашкова рекурсия, ефективно избиране на разделител, техники за избягване на лошия случай.

12. Сортиране чрез трансформация. Сортиране с линейна времева сложност. Сортиране чрез броене. Метод на бройните системи.
13. Съпоставка на алгоритмите за сортиране чрез сравняване и алгоритмите за сортиране чрез трансформация.
14. Търсене — същност и приложения. Последователно търсене. Двоично търсене. Бързо търсене. Детерминиран алгоритъм за търсене на k-ти най-малък / най-голям елемент в линейно време.
15. Алчни алгоритми.
16. Моделиране на практически задачи чрез графи. Представяния на граф — матрица на съседство, списъци на съседство; сравнение на двете представяния. Сложност на алгоритми върху графи — асимптотични нотации за функции на две променливи.
17. Обхождане на граф в ширина и в дълбочина. Видове ребра. Дърво на обхождането. Анализ на сложността. Приложения — алгоритми с линейна сложност: компоненти на силна и слаба свързаност, проверка за двуделност, търсене на цикли и др.
18. Ориентирани ациклични графи. Топологично сортиране.
19. Срязващи върхове и срязващи ребра (мостове) в неориентиран граф. Двусвързани компоненти на неориентиран граф. Търсене на срязващи върхове и ребра с линейна сложност по време.
20. Минимално покриващо дърво. Приложения. Обща алгоритмична схема за построяване на МПД. Алгоритъм на Прим—Ярник. Сравняване на реализациите — със или без приоритетна опашка, реализирана с двоична пирамида или с пирамида на Фибоначи.
21. Алгоритъм на Крускал за построяване на минимално покриващо дърво. Наивна реализация. Подобрена реализация — структура Union-Find, евристики Union by rank и компресиране на пътища.
22. Най-къс път в граф. Приложения в практически задачи. Търсене на най-къс път в нетегловен граф чрез обхождане в ширина. Алгоритъм на Дейкстра за граф с неотрицателни тегла на ребрата. Проблеми при графи с отрицателни тегла на ребрата.
23. Най-къси пътища в граф, съдържащ ребра с отрицателни тегла. Алгоритъм на Белман—Форд. Алгоритъм на Флойд—Уоршал.
24. Динамично програмиране — свеждане на задача с даден размер към задача с по-малки размери; мемоизация. Принцип на Белман за оптималност. Приложения на динамичното програмиране: търсене на най-къс път и брой на пътищата в ориентиран ацикличен граф, търсене на най-дълга растяща подредица, търсене на най-дълга обща подредица, търсене на подмножество от числа с даден сбор, разбиване на множество от числа на две подмножества с равни сборове, задача за раницата, синтактичен анализ (безконтекстни граматики в нормална форма на Чомски).

25. Паралелни алгоритми. Алгоритъм на Борувка за построяване на минимално покриващо дърво.
26. Сложност на изчислителна задача — сложност по време и по памет. Горни и долни граници на сложността на задача. Тривиални долни граници — по размера на входа и по размера на изхода. Методи за доказване на нетривиални долни граници: индукция, рекурентни неравенства, дърво за взимане на решения, игра срещу противник, редукция. Долни граници на сложността по време на конкретни задачи — сортиране чрез сравняване, проверка за повтарящи се елементи, най-малко разстояние, търсене на мода.
27. Практически решими изчислителни задачи — клас на сложност **P**. Практически нерешими изчислителни задачи. Примери върху неориентирани графи: върхово покритие, независимо множество, доминиращо множество. Практически нерешими задачи с кратък сертификат — клас на сложност **NP**.
28. Недетерминирани алгоритми за решаване в полиномиално време на задачи с кратък сертификат. Проблемът $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$. Класове на сложност **co-P** и **co-NP**; съпоставка.
29. Полиномиални редукции между изчислителни задачи. **NP**-пълни задачи. Теорема на Кук за **NP**-пълнотата на задачата SAT.
30. Изчислителната задача 3-SAT; доказателство за **NP**-пълнота. Съпоставка на 3-SAT и 2-SAT; доказателство, че 2-SAT $\in$ **P**. Доказателство на **NP**-пълнотата на някои задачи върху графи и върху масиви.
31. Апроксимацията като средство за справяне с практическата нерешимост. Примери за апроксимиращи алгоритми с фиксирана относителна точност — върхово покритие и задачата за търговския пътник в планарен вариант. Условна невъзможност за добро апроксимиране на задачи (при условие че $\mathbf{P} \neq \mathbf{NP}$).
32. Обобщения на понятието алгоритъм. Разширения на множеството от допустими алгоритми. Видове алгоритми — детерминирани и рандомизирани алгоритми (тип Монте Карло и тип Лас Вегас); евристични алгоритми; генетични / еволюционни алгоритми; алгоритми за квантов компютър.

Преподавател: гл. ас. д-р Д. Кралчев