

СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

учебна година: 2009/2010

семестър: летен
(зимен, летен)

наименование на дисциплината: Дизайн и анализ на алгоритми		
хорариум: 60 (45+15)	вид на дисциплината: задължителна (задължителна или избираема)	
Специалност: Компютърни науки	курс: втори	поток: първи
лектор: Минко Марков		

Приложения

1. Кратка анотация на дисциплината
2. Форма на проверка на знанията и уменията и начин на формиране на оценката по дисциплината
3. Тематичен план (конспект) на дисциплината
4. Литература

АНОТАЦИЯ

Частта **анализ** въвежда понятията *големина на входа*, *сложност по време* и *асимптотични граници на степента на нарастване*. Освен сложността в най-лошия случай се засяга и анализирането на средната сложност. Илюстрира се как използването на ефикасни структури данни подобрява сложността по време. Разглеждат се техниките за оценка на сложността на алгоритми и се демонстрира използването на тези техники върху класически образци. Разглеждат се техники за оценка на сложността на алгоритми и се демонстрира използването на тези техники върху класически образци. Въвежда се понятието *амортизиран анализ*. Въвежда понятието за *битова сложност*. Въвеждат се понятията *NP-пълнота*, *полиномиални редукции*. Анализират се няколко апроксимиращи алгоритъма като времева сложност и като отдалеченост на решението от оптималното.

В частта **„дизайн“** се въвеждат редица алгоритмични схеми, предназначени да улеснят създаването на ефикасни алгоритми: „разделяй и владей“, „динамично програмиране“, „greedy“, „обхождане на графи“, „backtracking“. Прилагат се схемите за решаване на типични задачи, като се демонстрират техниките за оценяване на сложност на получените алгоритми. Демонстрират се схеми за съставяне на апроксимиращи алгоритми.

ФОРМИРАНЕ НА ОЦЕНКАТА ПО ДИСЦИПЛИНАТА

	% от оценката
Текуща оценка	
- Първо контролно (теория): 3 ч.	= 45%
- Второ контролно (комп. състезание): 5 ч.	= 45%
- Домашни работи	= 5%
- Оценка на асистента	= 5%
Поправителен изпит (при текуща оценка под 3.00)	
- Въпрос от конспекта + задачи	= 50 %
- Практически (комп. състезание)	= 50 %

КОНСПЕКТ

1. Алгоритми – неформално определение и големина на входа. Сложност по време в най-лошия случай, средна и в най-добрия случай. Други аспекти на анализирането на алгоритмите – сложност по памет и коректност. Асимптотичен анализ на сложността – предимства и недостатъци. Нотации, използвани при асимптотичния анализ: O , o , Θ , Ω , ω .
2. Сортиране – приложения, важност, примери за използване на сортирането като първа фаза от решението. Стабилни и нестабилни сортиращи алгоритми. Двоични пирамиди. Пирамидална сортировка. Приоритетни опашки.
3. Рекурсивни алгоритми и рекурентни отношения. Методи за решаване на рекурентни отношения: развиване, дърво на рекурсията, индукция, Master Theorem, методът с характеристикното уравнение. Примери.
4. Алгоритмична схема „Разделяй и Владей“: същност, приложение и ограничения върху възможностите за прилагането ѝ. Mergesort и Quicksort. Анализ на коректността и сложността по време на двата алгоритъма.
5. Оценка на средната сложност по време на Quicksort. Долни граници върху асимптотичната сложност по време. Долна граница върху асимптотичната сложност по време на сортиращи алгоритми, базирани на директно сравнение. Сортиране в линейно време.
6. Графите от алгоритмична гледна точка. Представяния на графи чрез матрици на съседства и чрез списъци на съседства. Сравнителен анализ на предимствата и недостатъците на тези представяния. Основни алгоритми върху графи: търсене в ширина и търсене в дълбочина.
7. Алгоритми за топологично сортиране и силно свързани компоненти. Срязващи върхове на графи. Двусвързани компоненти на графи. Минимални покриващи дървета – алгоритми на Kruskal и Prim, анализ на сложността.
8. Най-къси пътища в графи. Алгоритъм на Dijkstra. Съпоставка между задачите за най-къс и най-дълъг път в граф. Най-дълъг път в ограничени класове графи – дървета и кактуси.
9. Динамично програмиране – същност, предимства и приложения. Примери: най-дълга обща подпоследователност, оптимално верижно умножение на матрици. Мемоизация – същност и сравнение с динамичното програмиране. „Лакоми“ алгоритми. Backtracking.
10. Комбинаторни алгоритми. Амортизиран анализ на сложността по време при тези алгоритми. Константното амортизирано време като оптимална времева сложност. Алгоритми за генериране на пермутации и комбинации. Ранкиране и деранкиране.
11. * Бързо умножение на числа и полиноми. Битова сложност. Бързо преобразуване на Фурие и дискретно преобразуване на Фурие. Бързо умножение на матрици.
12. Алгоритми за откриване на даден подстринг в стринг (string matching). Наивен алгоритъм. Алгоритъм на Knuth-Morris-Pratt – приложение на крайни детерминирани автомати. Анализ на времевата сложност.
13. * Масови задачи и сложност на задачи. Схеми за кодиране. Решаването на масови задачи като разпознаване на формален език. Класове на сложност P , NP и NP -с. Доказване на NP -пълнота (Теорема на Cook). Важност.

14. * Полиномиални редуции и Тюринг редуции. Подходи за доказване на NP -пълнота. Апроксимиращи алгоритми – примери с Vertex Cover и задачата за търговския пътник в „планарен вариант“.

Забележка: въпросите, отбелязани с „“, се четат само ако остане време!*

ЛИТЕРАТУРА

1. Кр. Манев, *Увод в дискретната математика*, IV издание, КЛМН, София, 2005г.
2. Кр. Манев, първа глава от учебник теория на алгоритмите (в процес на съставяне).
3. G. Brassard, P. Bratley, *Fundamentals of Algorithmics*, Prentice Hall, Englewood Cliffs, 1996.
4. T. Cormen, Ch. Leiserson, R. Rivest, *Introduction to Algorithms*, MIT Press, 1990.
5. M. Garey, D. Johnson, *Computers and Intractability. A Guide to the Theory of NP-c Completeness*, W.H. Freeman & Co., San Francisco, 1979.
6. S. Dasguta, C. H. Papadimitriou, U. Vazirani, *Algorithms*, McGraw-Hill Science/Engineering/Math, first edition, 2006. Книгата е достъпна безплатно и свободно на адрес <http://www.cs.berkeley.edu/~vazirani/algorithms/all.pdf>.
7. R. Graham, D. Knuth, O. Patashnik, *Concrete Mathematics. A Foundation for Computer Science*, Second Edition, Addison-Wesley, Reading, 1998.
8. Dan Gusfield, *Algorithms on Strings, Trees and Sequences*, Cambridge Univ. Press, 1997.
9. D. Knuth, *The Art of Computer Programming, vol.1 Fundamental Algorithms*, second edition, Addison-Wesley, 1973
10. D. Knuth, *The Art of Computer Programming, vol.3 Sorting and Searching*, second edition, Addison-Wesley, 1998
11. I. Parberry, *Problems on Algorithms*, Prentice Hall, 1995. Книгата е предоставена за безплатен достъп от автора, но само след приемането на условията от лиценза, на адрес <http://www.eng.unt.edu/ian/books/free/>.
12. I. Parberry, *Lecture Notes on Algorithm Analysis and Complexity Theory*, fourth edition, 2001. Книгата е предоставена за безплатен достъп от автора, но само след приемането на условията от лиценза, на адрес <http://www.eng.unt.edu/ian/books/free/>.
13. E. Reingold, J. Nievergelt, N. Deo, *Combinatorial algorithms. Theory and Practice*, Prentice Hall, Englewood Cliffs, 1977.
14. Fr. Ruskey, *Combinatorial Generation*, web draft, 2003. Книгата е достъпна безплатно и свободно на адрес <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.5967&rep=rep1&type=pdf>
15. R. Sedgewick, *Algorithms in C*, Addison-Wesley, Reading, 1990.
16. S. Skiena, *The Algorithm Design Manual*, Springer-Verlag, 1998.
17. S. Skiena, лекционни записи в аудио и видеоформат, както и слайдовете на презентациите, достъпни безплатно и свободно на адрес <http://www.cs.sunysb.edu/~algorithm/video-lectures/>.
18. H. Wilf, *Algorithms and Complexity*, Prentice-Hall, 1986. Книгата е достъпна безплатно и свободно на адрес <http://www.math.upenn.edu/~wilf/AlgoComp.pdf>.