СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛ. ОХРИДСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

КОНСПЕКТ

ЗА

ДЪРЖАВЕН ИЗПИТ ЗА ЗАВЪРШВАНЕ НА ОБРАЗОВАТЕЛНО-КВАЛИФИКАЦИОННА СТЕПЕН „БАКАЛАВЪР“

СПЕЦИАЛНОСТ

„КОМПЮТЪРНИ НАУКИ“

Промените в конспектите за ДИ са приети с
решение на ФС – Протокол № 05/29.05.2017 г.

СОФИЯ • 2018

При явяване на държавен изпит всеки студент е длъжен да носи студентската си книжка, да се яви навреме пред предварително оповестената зала и да спази указанията на квесторите за настаняване в залата.

Държавният изпит по специалност *„Компютърни науки“*е в две части, които се провеждат в два дни. През първия ден изпитът е практически (решаване на задачи) с продължителност 3 астрономически часа. Към въпроси с номера от 1 до 8, 11, 12, от 14 до 21 и от 27 до 31, могат да бъдат дадени задачи. Втория ден изпитът е теоретичен. Изтегля се един въпрос, който се развива за 2 астрономически часа. Работите се предават и се прави кратка почивка. Тегли се втори въпрос, който също се развива за 2 часа.

По време на всяка една част от изпита листата за писане са осигурени и подпечатани от ФМИ, други не се внасят. Пише се само с химикал - задължително син или черен цвят. Молив може да се използва само за чертежи.

По време на изпита може да се използва официално издадено копие на конспекта (получава се от квесторите). Всички други пособия са забранени.

Забранено е използването на електронни устройства от всякакъв вид. Необходимо е всички внесени мобилни устройства и компютърна техника да бъдат изключени преди започване на изпита и да бъдат оставени на определените за целта места. Намирането при студентите на нерегламентирани помощни средства се счита за опит за преписване. По време на изпита не се водят разговори, не се пуши и не се излиза от залата.

Работите се оценяват от комисия. Практическият и теоретичният изпит се оценяват поотделно. При положение, че и на двата изпита оценката е по-голяма или равна на 3.00, то крайната оценка от държавния изпит е закръглената по правилата средно аритметична оценка от двата изпита. В противен случай оценката е слаб (2.00). Оценката се закръгля до втори знак след десетичната запетая. Оценките са окончателни и не подлежат на преразглеждане.

Според правилника на СУ студентите нямат право на явяване за повишаване на оценка от ДИ, ако той е бил успешно положен. Напомняме на студентите, че според ЗВО за продължаване на образованието в ОКС „Магистър“ **(срещу заплащане)** е необходима оценка най-малко „добър“ от дипломата за ОКС „Бакалавър“.

КОНСПЕКТ ЗА ДЪРЖАВЕН ИЗПИТ
ЗА СПЕЦИАЛНОСТ “КОМПЮТЪРНИ НАУКИ”

ОСНОВИ НА КОМПЮТЪРНИТЕ НАУКИ

1. Множества. Декартово произведение. Релации. Функции.
2. Основни комбинаторни принципи и конфигурации. Рекурентни уравнения.
3. Графи. Дървета. Обхождания на графи.
4. Крайни автомати. Регулярни езици.
5. Контекстносвободни граматики и езици. Стекови автомати.
6. Сложност на алгоритъм. Асимптотично поведение на целочислени функции (О-, Ω-, Θ-, о- и ω-нотация). Сложност на рекурсивни програми.
7. Алгоритми в графи с тегла на ребрата. Оценки за сложност.
8. Динамично програмиране. Оценки за сложност.

ЯДРО НА КОМПЮТЪРНИТЕ НАУКИ

9. Компютърни архитектури. Формати на данните. Вътрешна структура на централен процесор – блокове и конвейрна обработка, инструкции.
10. Структура и йерархия на паметта. Сегментна и странична преадресация. Система за прекъсване – приоритети и обслужване.
11. Файлова система. Логическа организация и физическо представяне.
12. Управление на процеси и междупроцесни комуникации.
13. Компютърни мрежи и протоколи – OSI модел. Маршрутизация. Протоколи IPv4, IPv6, TCP, DNS.
14. Процедурно програмиране - основни информационни и алгоритмични структури.
15. Обектно ориентирано програмиране. Основни принципи. Класове и обекти. Наследяване и инкапсулация. Параметричен полиморфизъм.
16. Структури от данни. Стек, опашка, списък, кореново дърво. Основни операции върху тях. Реализация.
17. Обща характеристика на функционалния стил на програмиране. Дефиниране и използване на функции. Модели на оценяване. Функции от по-висок ред. Списъци. Потоци и отложено оценяване.
18. Синтаксис и семантика на термовете и формулите на предикатното смятане от първи ред. Унификация.
19. Метод на резолюцията в съждителното и в предикатното смятане от първи ред. Хорнови клаузи.
20. Бази от данни. Релационен модел на данните.
21. Бази от данни. Нормални форми.
22. Търсене в пространството от състояния. Генетични алгоритми.
23. Съвременни софтуерни технологии.
24. Архитектури на софтуерни системи.

МАТЕМАТИКА И ПРИЛОЖЕНИЯ

25. Симетрични оператори в крайномерни евклидови пространства. Основни свойства. Теорема за диагонализация.
26. Симетрична и алтернативна група. Теорема на Кейли. Теорема за хомоморфизмите на групи.
27. Теореми за средните стойности (Рол, Лагранж и Коши). Формула на Тейлър.
28. Определен интеграл. Дефиниция и свойства. Интегруемост на непрекъснати функции. Теорема на Нютон-Лайбниц.
29. Уравнения на права и равнина. Формули за разстояния.
30. Итерационни методи за решаване на нелинейни уравнения
31. Дискретни разпределения. Равномерно, биномно, геометрично и Поасоново разпределение. Задачи, в които възникват. Моменти – математическо очакване и дисперсия.

АНОТАЦИИ НА ВЪПРОСИТЕ

1. Множества. Декартово произведение. Релации. Функции.

Аксиоматизация на множествата – аксиоми за обема, отделянето, степенното множество и индуктивно генерираните множества. Математическа индукция. Основни операции върху множества и техните свойства. Наредена двойка и наредена n-орка. Декартово произведение и обобщено Декартово произведение на множества. Релация над n домейна. Свойства на бинарните релации. Релации на еквивалентост и класове на еквивалентност. Релации на частична наредба. Диаграми на Хассе. Релации на пълна наредба. Минимален и максимален елемент в релация на частична наредба. Влагане на частична наредба в пълна наредба – топологично сортиране. Частични и тотални функции. Инекции, биекции и сюрекции. Дефиниция на крайно множество и на кардиналност на крайно множество. Дефиниция на изброимо безкрайно множество. Принцип на Дирихле.

Примерни задачи

1. Да се докаже, че множествата, описани от дадени изрази, са равни: било чрез таблица, било чрез използване на свойствата на операции върху множества.
2. Да се докаже или опровергае, че дадена бинарна релация има дадено свойство.
3. Да се определят класовете на еквивалентност на дадена релация на еквивалентност.
4. Да се докаже по индукция дадено твърдение върху естествените числа.
5. Да се докаже твърдение чрез принципа на Дирихле.

Литература: [12], [30], [36].

2. Основни комбинаторни принципи и конфигурации. Рекурентни уравнения.

Формулировки на принципите на изброителната комбинаторика – принцип на Дирихле, принцип на биекцията, принцип на събирането, принцип на изваждането, принцип на умножението, принцип на делението, принцип за включване и изключване (с доказателство). Основните комбинаторни конфигурации: с или без наредба, с или без повтаряне. Извеждане на формулите за броя на основните комбинаторни конфигурации. Биномни коефициенти и теорема на Нютон. Доказателства на комбинаторни тъждества чрез комбинаторни разсъждения (принцип на двукратното броене). Алгоритъм за решаване на линейни рекурентни уравнения с константни коефициенти – хомогенни и нехомогенни.

Примерни задачи

1. Да се намери броят на частичните функции, на тоталните функциите, на инекции и на сюрекциите от крайно множество A в крайно множество B.

2. Да се намери броят на различните целочислени решения на уравнение от вида $x_1 + x_2 + \dots + x_n = M$, така че всички x_i са неотрицателни и трябва да удовлетворяват различни други условия, например $x_i \geq c_i$.
3. Да се реши зададено линейно (хомогенно или нехомогенно) рекурентно уравнение с константни коефициенти.

Литература: [12], [30], [36].

3. Графи. Дървета. Обхождания на графи.

Дефиниции за краен ориентиран (мулти)граф и краен неориентиран (мулти)граф. Дефиниции за път (цикъл) в ориентиран и неориентиран мултиграф. Свързаност и свързани компоненти на граф. Дефиниция на дърво и кореново дърво. Доказателство, че всяко кореново дърво е дърво и $|V|=|E|+1$. Покриващо дърво на граф. Обхождане на граф в ширина и дълбочина. Ойлерови обхождания на мултиграф. Теореми за съществуване на Ойлеров цикъл (с доказателство) и Ойлеров път.

Примерни задачи

1. Да се построи покриващо дърво на зададен граф – в ширина или дълбочина.
2. Да се построи Ойлеров цикъл (или път) в зададен мултиграф или да се докаже, че такъв цикъл (или път) не съществува.
3. Да се разбие множеството на ребрата на неориентиран граф на минимален брой пътища, никои два от които нямат общо ребро.

Литература: [12], [30], [36].

4. Крайни автомати. Регулярни езици.

Детерминирани крайни автомати. Недетерминирани крайни автомати. Представяне на всеки недетерминиран краен автомат с детерминиран (с доказателство). Регулярни операции. Доказателство за затвореност на автоматните езици относно регулярните операции. Регулярни езици. Формулировка и доказателство на теоремата на Клини. Формулировка и доказателство на лемата разрастване за регулярни езици (*uvw-лема*). Примери за нерегулярни езици. Формулировка и доказателство на теоремата на Майхил - Нероуд. Алгоритъм за конструиране на минимален краен детерминиран тотален автомат, еквивалентен на даден детерминиран краен автомат.

Примерни задачи

1. Да се построи детерминиран автомат, разпознаващ даден регулярен език.
2. По даден краен автомат да се намери еквивалентен на него минимален детерминиран краен автомат.
3. Да се докаже, че даден език е регулярен.
4. Да се докаже, че даден език не е регулярен.
5. Да се определи дали даден език е регулярен. Да се аргументира отговорът.

Литература: [12], [31], [33].

5. Контекстносвободни граматики и езици. Стекови автомати.
Контекстносвободна граматика, дърво на синтактичен анализ, контекстносвободен език. Доказателство на теоремите за затвореност на контекстносвободните езици.
Недетерминиран стеков автомат. Изпълнение в недетерминиран стеков автомат. Език, разпознаван от недетерминиран стеков автомат. Доказателство на теорема за свеждане на контекстносвободна граматика към еквивалентен недетерминиран стеков автомат.
Формулировка и доказателство на лемата за разрастване за контекстносвободни езици. Примери с доказателства за езици, които не са контекстносвободни. Доказателство за незатвореността на контекстносвободните езици относно допълнение и сечение.
Примерни задачи
1. Да се построи контекстносвободна граматика, пораждаща даден език.
2. По дадена контекстносвободна граматика да се построи стеков автомат, разпознаващ езика, породен от граматиката.
3. Да се докаже, че даден език е контекстносвободен.
4. Да се докаже, че даден език не е контекстносвободен.
5. Да се определи дали даден език е контекстносвободен. Да се аргументира отговорът.

Литература: [12], [31], [33].

6. Сложност на алгоритъм. Асимптотично поведение на целочислени функции (O-, Ω-, Θ-, o- и ω-нотация). Сложност на рекурсивни програми.
Модели на изчисленията – машина на Тюринг, машина с произволен достъп и език за програмиране. Дефиниции на (машинно-зависима) сложност (по време и памет) в най-лошия и средния случай. Поведение на асимптотически положителни целочислени функции – O-, Ω-, Θ-, o- и ω-нотация. Свойства и гранични теореми (без доказателство). Формулировка на теоремата за решенията на рекурентни отношения от вида $T(1)=\Theta(1)$, $T(n)=a.T(n/b) + f(n), n>1$.
Примерни задачи:
1. По зададено асимптотически положителна функция да се определи дали порядъкът ѝ на растеж е в зададено отношение (O-, Ω-, Θ-, o- или ω-) с порядъка на растеж на друга зададена асимптотически положителна функция.
2. По зададени две асимптотически положителни функции да се намери колкото може по-точно съотношението им на растеж една спрямо друга (в термините на O-, Ω-, Θ-, o- и ω-нотация) .
3. Да се реши зададено рекурентно отношение от вида $T(1)= \Theta(1)$, $T(n)=a.T(n/b) + f(n), n>1$, където $f(n)$ е мултипликативна.
4. Да се реши зададено рекурентно отношение от вида $T(1)= \Theta(1)$, $T(n)=a.T(n/b) + f(n), n>1$, където $f(n)$ не е мултипликативна.

Литература: [12], [28].

7. Алгоритми в графи с тегла на ребрата. Оценки за сложност.
Дефиниция на минимално покриващо дърво (МПД) на свързан граф с тегла на ребрата. Формулировка и доказателство на теоремата за съгласуваното множество (условия за нарастване на подмножество на МПД). Алгоритми на Прим и Крускал, имплементации и оценка на сложността. Задачи за най-къс път в граф с тегла на ребрата. Дърво на най-късите пътища. Алгоритъм за намиране на дърво на най-къси пътища в граф с константни тегла по ребрата и алгоритъм на Дейкстра. Оценка на сложността. Алгоритъм на Флойд за намиране на всички двойки най-кратки пътища.
Примерни задачи: задачи за програмиране, съответни на съдържанието на въпроса.
Литература: [12], [28].

8. Динамично програмиране. Оценки за сложност.
Същност на алгоритмичната схема „динамично програмиране" – свеждане на задача със зададен размер към задачи от същия вид с по-малки размери и „мемоизация". Принцип за оптималност и конструиране на решението на задачата от решенията на подзадачите. Задачи с линейна таблица на подзадачите (най-дълга растяща подредица). Задачи с триъгълна таблица на подзадачите (оптимално разбиване на редица). Задачи с правоъгълна таблица на подзадачите (най-дълга обща подредица на две редици, задача за раницата).
Примерни задачи: задачи за програмиране, съответни на съдържанието на въпроса.
Литература: [28].

9. Компютърни архитектури – Формати на данните. Вътрешна структура на централен процесор – блокове и конвейерна обработка, инструкции.
Обща структура на компютрите и концептуално изпълнение на инструкциите, запомнена програма. Формати на данните – цели двоични числа, двоично-десетични числа, двоични числа с плаваща запетая, знакови данни и кодови таблици. Централен процесор – регистри, АЛУ, регистри на състоянията и флаговете, блокове за управление, връзка с паметта, дешифрация на инструкциите, преходи.
Литература: [5], [38], [43].

10. Структура и йерархия на паметта. Сегментна и странична преадресация. Система за прекъсване – приоритети и обслужване.
Структура на основната памет. Йерархия – кеш, основна и виртуална памет. Сегметна и странична преадресация – селектор, дескриптор, таблици и регистри при сегметна преадресация; каталог на страниците, описател, стратегии на подмяна на страниците при странична преадресация. Система за прекъсване – видове прекъсвания, структура и обработка, конкурентност и приоритети, контролери на прекъсванията.
Литература: [5], [38], [43].

11. Файлова система. Логическа организация и физическо представяне.
Логическа организация на файлова система (ФС). Имена на файлове. Типове файлове - обикновен файл, специален файл, каталог, символна връзка, програмен канал. Вътрешна структура на файл. Атрибути на файл. Йерархична организация на ФС - абсолютно и относително пълно име на файл, текущ каталог. Физическа организация на ФС. Стратегии за управление на дисковото пространство. Системни структури, съдържащи информация за разпределението на дисковата памет и съхранявани постоянно на диска: за свободните блокове; за блоковете, разпределени за всеки един файл; за общи параметри на ФС. Примери за физическа организация на ФС: UNIX System V; LINUX; MS DOS; NTFS.
Забележки: За изпита ще бъдат избрани два от изброените примери за файлова организация.
Примерни задачи: задачи, съответни на съдържанието на въпроса.
Литература: [14], [42].

12. Управление на процеси и междупроцесни комуникации.
Основни системни примитиви за управление на процеси. Създаване на процес. Изпълнение на програма. Завършване на процес. Синхронизация със завършването на процеса-син. Права на процеси – потребителски идентификатори на процес. Групи процеси и сесия. Механизми за междупроцесни комуникации. Сигнали. Програмни канали. IPC пакет на UNIX System V: Обща памет. Семафори. Съобщения.
Примерни задачи: задачи, съответни на съдържанието на въпроса.
Литература: [22], [39].

13. Компютърни мрежи и протоколи – OSI модел. Маршрутизация. Протоколи IPv4, IPv6, TCP, DNS.
OSI модел – най-обща характеристика на нивата, съпоставяне с модела TCP/IP. Разпределена маршрутизация – алгоритми с дистантен вектор и следене на състоянието на линията. IPv4 адресация – класова и безкласова. Основни характеристики на протокол IPv6. TCP – процедура на трикратно договаряне. Основни характеристики на протоколи DNS (резолвинг на имената по IPv4 и IPv6).
Литература:[3], [32], [44].

14. Процедурно програмиране - основни информационни и алгоритмични структури
Принципи на структурното програмиране. Управление на изчислителния процес. Основни управляващи конструкции – условни оператори, оператори за цикъл. Променливи – видове: локални променливи, глобални променливи; инициализация на променлива; оператор за присвояване. Функции и процедури. Параметри – видове параметри. Предаване на параметри – по име (по указател), по стойност. Типове и проверка за съответствие на тип. Типовете като параметри на функция. Едномерни и многомерни масиви. Основни операции с масиви – индексиране. Сортиране и търсене в едномерен масив – основни алгоритми за

сортировка. Рекурсия – пряка и косвена рекурсия, линейна и разклонена рекурсия.
Литература: [24], [40] или [26], [46].

15. Обектно ориентирано програмиране. Основни принципи. Класове и обекти. Наследяване и инкапсулация. Параметричен полиморфизъм.
Обекти (екземпляри). Класове. Декларация на клас и декларация на обект. Конструктори – конструктор по подразбиране, конструктор за присвояване. Освобождаване на памет – процедура за събиране на свободната памет. Методи – декларация, предаване на параметри, връщане на резултат. Наследяване и достъп до наследените компоненти. Еднократно и многократно (множествено) наследяване. Производни и вложени класове. Абстрактни методи и класове. Инкапсулация и скриване на информацията. Модификатори – статични полета и методи. Параметричен полиморфизъм. Типовете като параметри към функция и клас.
Примерни задачи: задачи за програмиране, съответни на съдържанието на въпроса.
Литература: [24], [40] или [26], [46].

16. Структури от данни. Стек, опашка, списък, кореново дърво. Основни операции върху тях. Реализация.
Структури от данни - дефиниране. Линейни структури от данни – списък, опашка, стек. Логическо описание. Статични и динамични реализации. Дефиниране на класове, реализиращи статично или динамично свързан списък (едностранно или двустранно), опашка или стек. Дървовидни структури от данни – двоично кореново дърво и двоично кореново дърво за бързо търсене. Логическо описание. Статични и динамични реализации. Дефиниране на класове, реализиращи двоично кореново дърво или двоично кореново дърво за бързо търсене.
Забележка. За изпита ще бъдат избрани една от линейните и една от дървовидните структури.
Примерни задачи: задачи за програмиране, съответни на съдържанието на въпроса.
Литература: [25], [11], [16].

17. Обща характеристика на функционалния стил на програмиране. Дефиниране и използване на функции. Модели на оценяване. Функции от по-висок ред. Списъци. Потоци и отложено оценяване.
1. Характерни особености на функционалния стил на програмиране. Основни компоненти на функционалните програми. Примитивни изрази. Средства за комбиниране и абстракция. Оценяване на изрази. Дефиниране на променливи и процедури. Среди. Специални форми. Модели на оценяване на комбинации. Процедури от по-висок ред. Процедурите като параметри и оценки на обръщения към процедури.
2. Списъци. Основни операции със списъци. Процедури от по-висок ред за работа със списъци. Потоци. Основни операции с потоци. Функции от по-висок ред за работа с потоци. Отложено оценяване. Работа с безкрайни потоци.
Забележка. На изпита ще бъде давана една от двете части на въпроса.

Литература: [15], [23].

18. Синтаксис и семантика на термовете и формулите на предикатното смятане от първи ред. Унификация.

Дефинират се синтактичните понятия терм и формула от даден език на предикатното смятане. Дефинират се понятията унификатор и най-общ унификатор за множество от термове. Формулира се алгоритъм за намиране на най-общ унификатор за крайно множество от термове.

Дава се семантика на термовете и формулите в дадена структура за езика. Доказва се, че множество от затворени универсални формули има модел точно тогава, когато множеството от частните му случаи е булево изпълнимо.

Литература: [13], [20], [34].

19. Метод на резолюцията в съждителното и в предикатното смятане от първи ред. Хорнови клаузи.

Дефинира се понятието съждителен резолютивен извод и се доказва теоремата за коректност и пълнота на резолютивната изводимост. Описва се метода на резолюцията. Дефинира се понятието хорнов дизюнкт и се доказва, че изпълнимите множества от хорнови дизюнкти имат най-малък модел.

Примерни типове задачи, свързани с въпроси 18 и 19:

1. Практически задачи – за дефиниране на предикат с помощта на Пролог; за проследяване на изпълнението на програма на Пролог.

2. Теоретични задачи – за определимост и неопределимост на свойства в дадена структура; показване на изпълнимост на множество от предикатни формули чрез посочване на структура; доказване на неизпълнимост на множество от предикатни формули с помощта на метода на резолюцията.

Литература: [13], [20], [34].

20. Бази от данни. Релационен модел на данните.

Релационен модел на данните: домейн; релация; кортежи; атрибути; схема на релация; схема на релационна база от данни; реализация на релационната база от данни; видове операции върху релационната база от данни; заявки към релационната база от данни. Релационна алгебра: основни (обединение; разлика; декартово произведение; проекция; селекция) и допълнителни (сечение; частно; съединение; естествено съединение) операции.

Примерни задачи: Съставяне на SQL заявки, DDL и DML команди.

Литература: [29].

21. Бази от данни. Нормални форми.

Нормални форми. Проектиране схемите на релационните бази от данни. Аномалии, ограничения, ключове. Функционални зависимости, аксиоми на Армстронг. Първа, втора, трета нормална форма, нормална форма на Бойс-Код. Многозначни зависимости; аксиоми на функционалните и многозначните зависимости; съединение без загуба; четвърта нормална форма.

Примерни задачи: Привеждане на схема на базата от данни (при зададени функционални зависимости) към зададена нормална форма.
Литература [29].

22. Търсене в пространството от състояния. Генетични алгоритми.
Пространство на състоянията. Основни понятия. Формулировка на основните типове задачи за търсене в пространството на състоянията: търсене на път до определена цел, формиране на стратегия при игри за двама играчи, намиране на цел при спазване на ограничителни условия. Методи за информирано (евристично) търсене на път до определена цел: best-first search, beam search, hill climbing, A*. Генетични алгоритми – основен алгоритъм, типове кръстосване и мутация.
Литература [41].

23. Съвременни софтуерни технологии.
Софтуерен продукт и процес. Модел на софтуерен процес. Софтуерни технологии. Управление на софтуерен проект и ресурсите. Участници в софтуерния процес. Функционални и нефункционални изисквания. Анализ и проектиране на софтуерните изсиквания. Проектиране на софтуера. Обектно-ориентиран дизайн. Езици за описание. UML. Верификация и валидация на софтуера. Тестване на софтуера. Управление на процеса на тестване. Управление на качеството на процеса на създаване на софтуер. Съвременни софтуерни технологии. Гъвкави (agile) софтуерни технологии. Extreme Programming (XP). Test Driven Development. Feature driven development. SCRUM.
Литература [9], [35], [37]

24. Архитектури на софтуерни системи.
Софтуерни технологии центрирани около софтуерната архитектура. Качествени атрибути – дизайн на архитектурата. Дизайн за постигане на ефективност, сложност, скалируемост и хетерогенност, адаптируемост на архитектурата. Надеждност и сигурност. Компоненти и конектори. Типове конектори и техните променливи характеристики. Критерий за избор на подходящи конектори. Архитектурни стилове. Разпределени, мрежови, децентрализирани архитектури. Архитектури, ориентирани към услуги и уеб услуги. Клиент –сървър. Анализ и визуализация на софтуерна архитектура.
Литература [27], [45]

25. Симетрични оператори в крайномерни евклидови пространства. Основни свойства. Теорема за диагонализация.
Определение за симетричен оператор. Матрица на симетричен оператор спрямо ортонормиран базис. Всички характеристични корени на симетричен оператор са реални числа. Всеки два собствени вектора, съответстващи на различни собствени стойности, са ортогонални помежду си. Съществува ортонормиран базис на пространството, в който матрицата на симетричен оператор е диагонална.

Литература: [19].

26. Симетрична и алтернативна група. Теорема на Кейли. Теорема за хомоморфизмите на групи.

Симетрична група S_n – представяне на елементите като произведение на независими цикли. Спрягане на елементите на S_n. Транспозиции и представяне на елементите като произведение на транспозиции. Алтернативна група. Теорема на Кейли – всяка крайна група е изоморфна на подгрупа на симетричната група. Хомоморфизъм при групи, ядро и образ. Теорема за хомоморфизмите при групи.
Литература: [18].

27. Теореми за средните стойности (Рол, Лагранж и Коши). Формула на Тейлър.

Необходимо е да се докажат следните, формулирани общо, теореми: Нека f е непрекъсната в затворения интервал $[a,b]$ и притежава производна поне в отворения интервал (a,b). Да се докаже, че:

а) ако $f(a)=f(b)$, то съществува такова $c\in(a,b)$, че $f'(c)=0$ (***Рол***);

б) съществува такова $c\in(a,b)$, че $f(b)-f(a)=f'(c)(b-a)$ (***Лагранж***);

в) ако g е непрекъсната в затворения интервал $[a,b]$ и притежава производна поне в отворения интервал (a,b), $g'(x)\neq 0, x\in(a,b)$, то съществува такова $c\in(a,b)$, че

$$\frac{f(b)-f(a)}{g(b)-g(a)}=\frac{f'(c)}{g'(c)} \qquad (\textbf{\textit{Коши}}).$$

За доказателството на теоремата на Рол да се използва (*без доказателство!)* теоремата на Вайерщрас, според която всяка непрекъсната функция в краен и затворен интервал достига своя максимум и минимум.
Необходимо е още да се изведе формулата на Тейлър с остатъчен член във формата на Лагранж.
Примерни задачи: Като се използва теоремата на Лагранж, да се докаже, че

$$ln(1+x)<x \text{ при } x>0$$

или, че

$$|arctg\, x_2 - arctg\, x_1| \leq |x_2 - x_1| \text{ за всяко } x_1, x_2 \in \mathbb{R}.$$

Литература: [6], [8], [10].

28. Определен интеграл. Дефиниция и свойства. Интегруемост на непрекъснатите функции. Теорема на Нютон - Лайбниц.

Да се дефинират последователно: разбиване на интервал, диаметър на разбиване, риманова сума и риманов интеграл. Да се покаже, че всяка интегруема по Риман функция е ограничена.
Да се дефинират големи и малки суми на Дарбу. Да се установи, че при добавяне на нови точки в разбиването на интервала, големите суми на Дарбу не нарастват, а малките не намаляват (*желателно е да се направи чертеж*).
Да се докаже, че дадена функция е интегруема по Риман тогава и само тогава, когато за всяко $\varepsilon > 0$ съществуват голяма сума на Дарбу S и малка сума на Дарбу s такива, че $S - s < \varepsilon$. Като се използва тази теорема и теоремата на Кантор, според която всяка непрекъсната функция в краен и затворен интервал е равномерно непрекъсната, да се докаже, че всяка непрекъсната функция в краен и затворен интервал е интегруема по Риман. Да се изброят (*без доказателство*) основните свойства на Римановия интеграл. Като се приложи свойството за интегриране на неравенства и теоремата, че всяка непрекъсната функция приема всички стойности между максимума и минимума си, да се докаже, че ако f е непрекъсната в $[a,b]$, то съществува $c \in [a,b]$ така , че

$$\int_a^b f(x)dx = f(c)(b-a).$$

Да се докаже теоремата на Нютон-Лайбниц, т.е. ако f е непрекъсната в $[a,b]$, то за всяко $x \in [a,b]$

$$\frac{d}{dx}\int_a^x f(t)dt = f(x);$$

да се покаже как теоремата се използва за изчисляване на определен интеграл.
Примерни задачи: Смяна на променливите и интегриране по части; интегриране на рационални функции; интеграли от вида

$$\int_c^d \frac{dx}{(x^2+a^2)^2};$$

интегриране на класове от ирационални функции, субституции на Ойлер; субституции за интегриране на рационални функции от $\sin x$ и $\cos x$.
Литература: [6], [8], [10].

29. Уравнения на права и равнина. Формули за разстояния.
Векторни и параметрични (скаларни) уравнения на права и равнина. Общо уравнение на права в равнината. Декартово уравнение. Взаимно положение на две прави. Нормално уравнение на права. Разстояние от точка до права. Общо уравнение на равнина. Взаимно положение на две равнини. Нормално уравнение на равнина. Разстояние от точка до равнина.
Литература: [21].

30. Итерационни методи за решаване на нелинейни уравнения.

Да се дефинира понятието *неподвижна точка* на изображението φ и да се докаже, че ако φ е непрекъснато изображение на интервала $[a,b]$ в себе си, то φ има поне една неподвижна точка в $[a,b]$. Да се покаже, че решаването на уравнението $f(x) = 0$ може да се сведе към намиране на неподвижна точка.

Да се дефинира понятието *свиващо изображение* и да се докаже, че ако φ е непрекъснато изображение на интервала $[a, b]$ в себе си и е свиващо с константа на Липшиц $q<1$, то: а) уравнението $x = \varphi(x)$ има единствен корен ξ в $[a,b]$; б) редицата $\{x_n\}$ от последователни приближения (при произволно $x_0 \in [a,b]$ и $x_{n+1} = \varphi(x_n)$, $n = 0,1,2,\ldots$, клони към ξ при $n \to \infty$, като $|x_n - \xi| \le (b-a)q^n$, за всяко n. Да се получи като следствие, че ако ξ е корен на уравнението $x = \varphi(x)$ и φ има непрекъсната производна в околност U на ξ, за която $|\varphi'(\xi)| < 1$, то при достатъчно добро начално приближение x_0 итерационният процес, породен от φ, е сходящ със скоростта на геометрична прогресия. Да се дефинира понятието *ред на сходимост*.

Да се дадат геометрична илюстрация, формула за последователните приближения и ред на сходимост при: метод на хордите, метод на секущите и метод на Нютон. Да се докаже, че при метода на хордите сходимостта е със скоростта на геометричната прогресия (при условие, че коренът е отделен в достатъчно малък интервал).

Литература: [1], [2], [17].

31. Дискретни разпределения. Равномерно, биномно, геометрично и Поасоново разпределение. Задачи, в които възникват. Моменти – математическо очакване и дисперсия.

На изпита комисията дава две разпределения, върху които се развива въпросът. Дефиниция на дискретно вероятностно разпределение на случайна величина. Свойства на вероятностите (неотрицателност и нормираност, монотонност и адитивност). За всяко от дадените две разпределения да се посочи пример, при който то възниква. Да се пресметне математическото очакване и дисперсията на всяко от тези разпределения. При пресмятанията може да се използва пораждаща функция или пораждаща моментите функция, но тя трябва да се дефинира и да се покажат основните й свойства (без доказателство).

Литература: [7], глави 2.3 (стр. 54-56), 3.2 (стр. 71-74), 6.1 (примери 1-4); [4], тема: Дискретни разпределения.

ЛИТЕРАТУРА

1. Андреев, А. и др., Сборник от задачи по числени методи, Университетско издателство “Св. Кл. Охридски”, София, 1994.
2. Боянов, Б., Лекции по числени методи, Дарба, София, 1998.

3. Боянов, Л., К. Боянов и др., Компютърни мрежи и телекомуникации, изд. "Авангард Прима", София, 2014.
4. Въндев, Д., Записки по теория на вероятностите, Електронно издание: http://fmi.uni-sofia.bg/fmi/statist/personal/vandev/lectures/prob/prob.htm
5. Горслайн, Дж., Фамилия Intel 8086/8088, Техника, София, 1990.
6. Джаков, П., Р. Леви, Р. Малеев, С. Троянски, Диференциално и интегрално смятане, ФМИ-СУ, София, 2004.
7. Димитров, Б., К. Янев, Вероятности и статистика, Университетско издателство "Св. Кл. Охридски", София, 1998.
8. Дойчинов, Д., Математически анализ, Университетско издателство "Св. Кл. Охридски", София, 1994.
9. Ескенази А., Манева Н.,Софтуерни технологии , 2006.
10. Любенова, Е., П. Недевски, К. Николов, Л. Николова, В. Попов, Ръководство по Математически анализ, София, 1998.
11. Майерс, С., По-ефективен С++. 35 нови начина да подобрите своите програми и проекти, ЗеСТ Прес.
12. Манев, К., Увод в дискретната математика. Издателство "КЛМН – Красимир Манев", София, пето редактирано издание, 2012, ISBN 954-535-136-5.
13. Метакидес, Д., А. Нероуд, Принципи на логиката и логическото програмиране, Виртех, София, 2000.
14. Николов, Л., Операционни системи, Сиела, София, 1998.
15. Нишева, М., П. Павлов, Функционално програмиране на езика Scheme, София, 2004.
16. Седжуик, Р., Алгоритми на С, ч.1-4: Основи, структури от данни, сортиране, търсене, СофтПрес
17. Сендов, Бл., В. Попов, Числени методи, I ч., Университетско издателство "Св. Кл. Охридски", София, 1996.
18. Сидеров, Пл., Чакърян, К., Записки по алгебра: групи, пръстени, полиноми, Веди, София, 2014.
19. Сидеров, Пл., Чакърян, К., Записки по алгебра: линейна алгебра, Веди, София, 2014.
20. Скордев, Д., Логическо програмиране (записки). Електронно издание: http://www.fmi.uni-sofia.bg/fmi/logic/skordev/ln/lp/new/sydyrzha.htm
21. Станилов, Гр., Аналитична геометрия, Софтех, София, 1998.
22. Стивенс, У. UNIX: взаимодействие процессов. СПб.: Питер, 2003.
23. Тодорова, М., Езици за функционално и логическо програмиране, I ч.: Функционално програмиране. Сиела, София, 2003.
24. Тодорова, М., Програмиране на С++, I и II част. Ciela, София, 2002.
25. Уирт, Н., Алгоритми + структури от данни = програми, BG soft group, София.
26. Bruce Eckel, Thinking in Java. Prentice Hall, 4th ed., 2006 (Да мислим на Java, превод и издание на български език).

27. Clements P., Bachmann F., Bass L., Garlan D., Ivers J., Little R., Merson P., Nord R., Stafford J., Documenting Software Architectures: Views and Beyond, Addison-Wesley Professional, 2010, ISBN: 0-321-55268-7
28. Cormen, T., Ch. Leiserson, R. Rivest, Introduction to Algorithms, MIT Press, 1990.
29. Garcia-Molina, H., J. Ullman, J. Widom, Database Systems: The Complete Book, Prentice Hall, 2002.
30. Grimaldi, R., Discrete and Combinatorial Mathematics: An Applied Introduction, Pearson, 5 edition, 2003, ISBN-13: 978-0201726343.
31. H. Lewis., Chr. Papadimitriou, Elements of the theory of computation., Second edition, Prentice-Hall, 1998.
32. Larry L. Peterson and Bruce S. Davie, Computer Networks: A Systems Approach Fifth Edition, © 2012 Elsevier, Inc.
33. Martin, J., Introduction to Languages and the Theory of Computation, McGraw-Hill, 4 edition, 2010, ISBN-13: 978-0073191461.
34. Nilsson, U., J. Maluszynski, Logic, Programming and Prolog (2nd ed.). John Willey & Sons, 1995. Електронно издание: http://www.ida.liu.se/~ulfni/lpp/
35. Pressman R., Maxim B., Software Engineering: A Practitioner's Approach, McGraw-Hill, 8/e, 2015, ISBN: 0078022126.
36. Rosen, K., Discrete Mathematics and Its Applications, McGraw-Hill Education, 7 edition, 2012, ISBN 9780073383095.
37. Sommerville I., Software Engineering, Addison-Wesley 10th ed., 2015.
38. Stallings, W., Computer Organization and Architecture. Design for Performance, Prentice Hall, 2000. http://www.williamstallings.com/COA5e.html
39. Stevens W.R. Advanced Programming in the UNIX Environment, Addison-Wesley, Reading, Mass, 1992.
40. Stroustrup, B., C++ Programming Lanquage. Third Edition, Addison-Wesley, 1997.
41. Stuart Russel & Peter Norvig, Artificial Intelligence: A Modern Approach. Prentice Hall, 2003.
42. Tanenbaum, A., Modern Operating systems, 2nd ed., Prentice Hall, 2002.
43. Tanenbaum, A., Structured Computer Organization. Prentice Hall, 2002.
44. Tannenbaum Andrew S., Wetherall David J., Computer Networks, 5th ed., Prentice Hall, 2011, http://libgen.org/book/index.php?md5=1990789686fa1463f09d2fb230d4301c
45. Taylor R., Medvidovic N., Dashofy E Software Architecture: Foundations, Theory, and Practice,., John Wiley & Sons, 2009, ISBN 0470167742.
46. Timothy Budd, An Introduction to Object-Oriented Programming. Addison Wesley, 3rd ed., 2002.