

Конспект за група С

Целта на този документ е да служи като набор от насоки, които да помогнат да се прецени дали дадена задача е подходяща за група С. Но не може да служи като строго ограничение за теми, които не са явно или косвено споменати. На базата на този документ отговорниците на група С оценяват предложените задачи за състезанията.

Конспектът, представен по-долу, е съставен с помощта на конспекта на Европейската младежка олимпиада по информатика (EJOI). Някои допълнителни теми (спрямо конспекта на EJOI) са отбелязани с +. Освен това с ~ са отбелязани теми, които могат да участват само в края на учебната година – след пролетните състезания.

1. Математика

1.1. Аритметика

- ✓ Цели числа, операции (включително степенуване), сравнение
- ✓ Дробни, проценти
- ✓ Основни свойства на целите числа (знак, четност, делимост)
- ✓ Прости числа
- ✓ НОД и НОК, основни свойства на НОД
- ✓ Основна модулна аритметика: събиране, изваждане, умножение
- ✓ По-сложна модулна аритметика: делене и обратни елементи (при прости модули)+~
- ✓ Малка теорема на Ферма+~
- ✓ Бройни системи
- ✗ Допълнителни теми от теория на числата
- ✗ По-сложен анализ за увеличаване на точността на изчисленията с десетични числа
- ✗ Комплексни числа

1.2. Геометрия

- ✓ Права, отсечка, ъгъл, триъгълник, правоъгълник, квадрат, окръжност
- ✓ Точка, вектор, координати в равнината
- ✓ Многоъгълник (върх, страна, прост, изпъкнал, вътрешност, лице)
- ✓ Евклидово разстояние
- ✓ Питагорова теорема
- ✗ Изпъкнала обвивка
- ✗ Геометрия в три или повече измерения
- ✗ Основни криви (параболи, хиперболи, елипси)
- ✗ Тригонометрични функции

1.3. Дискретни структури (ДС)

ДС1. Множества, релации и функции

Тази подсекция включва фундаментални знания, които са нужни за измислянето и/или доказването на решения на задачите, но ако конкретно понятие се използва в условието, то ще има изчерпателно определение.

- ✓ Множества (включване/изключване, допълнение, Декартово произведение, степенно множество)
- ✓ Релации (рефлексивност, симетричност, транзитивност, релация на еквивалентност, пълни/линейни наредби, лексикографска наредба)
- ✓ Функции (сюрекции, инекции, обратни, композиция)
- ✗ Кардиналност и изброимост (на безкрайни множества)

ДС2. Основна логика

- ✓ Логика от първи ред
- ✓ Логически връзки (включително техни основни свойства)
- ✓ Таблица за истинност
- ✓ Квантори за всеобщност и за съществуване
- ✓ Използване и прилагане на основни закони за импликация
- ✗ Нормални форми
- ✗ Валидност
- ✗ Ограничения на предикатната логика

ДС3. Техники за доказателства

- ✓ Понятие за импликация, обратно, противоположно, противопоставително твърдение, отрицание и противоречие
- ✓ Директни доказателства, доказателства с: контрапример, контрапозиция, противоречие
- ✓ Математическа индукция (и силна индукция)
- ✓ Рекурентни математически дефиниции (включително взаимно рекурентни дефиниции)

ДС4. Основи на броенето

- ✓ Правила за броене (правила за сбор и произведение, аритметични и геометрични прогресии, редицата на Фибоначи)
- ✓ Пермутации, вариации и комбинации с или без повторение (дефиниции и основни приложения)
- ✓ Факториел, биномни коефициенти
- ✓ Принцип на включването и изключването
- ✓ Принцип на Дирихле
- ✓ Триъгълник на Паскал, биномна теорема
- ✗ Решаване на рекурентни релации
- ✗ Лема на Бърнсайд

ДС5. Графи и дървета

- ✓ Неориентирани графи (върх/възел, ребро, степен, съседство)
- ✓ Ориентирани графи (полустепен на входа и изхода) и ациклични ориентирани графи (DAG)
- ✓ Мултиграфи, графи с примки
- ✓ Двуделни графи
- ✓ „Декорирани“ графи с етикети, тегла, цветове на ребрата/върховете
- ✓ Пътища в графи (неориентиран и ориентиран път, цикъл, Ойлеров път/цикъл, Хамилтонов път/цикъл)
- ✓ Достижимост (свързана компонента, най-кратък път)
- ✓ Дървета (листа, диаметър, гора)
- ✓ Коренови дървета (корен, родител, дете, предшественик, поддърво, двоични дървета)
- ✓ Покриващи дървета (и като цяло подграфи)
- ✓ Стратегии за обхождане
- ✓ Основни комбинаторни свойства на графите¹
- ✗ По-сложни свързаности в графи (двусвързаност и силно-свързани компоненти в ориентирани графи)
- ✗ Планарни графи
- ✗ Хиперграфи
- ✗ Специфични класове графи като перфектни графи

ДС6. Дискретна вероятност – ✗

1.4. Други области

- ✓ Основи на комбинаторната теория на игрите, печеливши и губещи позиции
- ✓ Матрици (дефиниция)
- ✓ Основни статистики като средно аритметично, медиана
- ✗ Допълнителна теория на комбинаторните игри (като играта Ним и теоремата на Спраг-Грънди)
- ✗ Линейна алгебра, включително (но не само):
 - Умножение на матрици, повдигане в степен, обратна матрица и Гаусова елиминация
 - Интерполация на полиноми
 - Бързо преобразование на Фурие
- ✗ Анализ
- ✗ Допълнителни теми от статистиката

¹Тази точка включва различни връзки между броя върхове, ребра и свързани компоненти в граф, както и за степени на върховете и други подобни свойства. Един пример е лемата за ръкостискането.

2. Информатика

2.1. Основи на програмирането (ОП)

ОП1. Основни програмни конструкции

- ✓ Основен синтаксис и семантика на C++
- ✓ Променливи, типове, изрази и присвояване
- ✓ Прост вход и изход
- ✓ Условни и итерационни контролни структури
- ✓ Функции и подаване на параметри
- ✓ Рекурсия
- ✓ Побитови операции
- ✓ Структуриране на кода и разделяне на логически части

ОП2. Основни структури от данни

- ✓ Примитивни типове (булев, целочислен с и без знак, символен)
- ✓ Масиви
- ✓ Низове и тяхната обработка
- ✓ Статична алокация в стека (елементарно автоматично управление на паметта)
- ✓ Свързани структури
- ✓ Имплементационни стратегии за графи и дървета
- ✓ Елементарна употреба на дробни числа
- ✓ Представянето на дробните числа като числа с плаваща запетая и съществуването на грешки при точността²
- ✓ Указатели и референции
- ✓ Представяне на данните в паметта
- ✗ Алокация в хийпа
- ✗ Управление на съхранението по време на изпълнение
- ✗ Нетривиални изчисления с числа с плаваща запетая, манипулиране на грешки при точността

ОП3. Четене от и писане в текстови файлове

Някои задачи могат да са от вид output-only, така че от състезателите може да се очаква да направят техните програми да четат данни от и да записват данни в текстови файлове според описан прост формат.

ОП4. Програмиране на база събития

Някои задачи могат да имат елементи на диалог с програмата на журито. Съответно при тях ще се включва имплементиране на подходяща интеракция с предоставената среда.

²Когато е възможно, очакването е да се избегнат сметките с плаваща запетая, както и да не е необходимо усилено използване на обикновени дроби, за да се сметне точният резултат.

2.2. Алгоритми и сложност (АЛ)

АЛ1. Анализ на алгоритми

- ✓ Спецификация на алгоритми, предусловия, следусловия, коректност, инварианти
- ✓ Асимптотичен анализ на горни граници за сложност
- ✓ Амортизационен анализ
- ✓ Нотация O -голямо
- ✓ Стандартни класове сложности: константна, логаритмична, линейна, $O(n \log n)$, квадратична, кубична, експоненциална и други
- ✓ Компромис между време и памет за алгоритмите
- ✓ Емпирично замерване на изпълнението
- ✓ Определяне на разликите между поведението в най-добрия, средния и най-лошия случай
- ✓ Регулиране на параметрите за намаляване на изпълнението по време, памет или друга мярка
- ✗ Асимптотичен анализ на граници за сложност в средния случай
- ✗ Използване на рекурентни релации за анализиране на рекурсивни алгоритми (с изключение на простото рекурентно отношение за анализ на сортиране чрез сливане)

АЛ2. Стратегии за алгоритми

- ✓ Прости циклични стратегии
- ✓ Пълно изчерпване
- ✓ Алчни алгоритми
- ✓ Разделяй и владей
- ✓ Търсене с връщане (рекурсивно и нерекурсивно). Метод на разклоняването и ограничаването
- ✓ Динамично програмиране, включително (но не само):
 - Основни и класически ДП
 - ДП с битови маски
 - ДП по цифри
 - ДП в дърво и ациклични ориентирани графи
- ✗ Среждане по средата
- ✗ Коренова декомпозиция (включително Алгоритъм на Мо)
- ✗ Евристики
- ✗ Дискретни апроксимационни алгоритми
- ✗ Рандомизирани алгоритми
- ✗ Намиране на добри характеристики за задачи с машинно самообучение
- ✗ Алгоритми за клъстеризиране (например k средни, k най-близки съседа)
- ✗ Минимизиране на функция на повече от една променлива, използвайки числени методи

АЛ3а. Прости алгоритми

- ✓ Прости алгоритми с числа: превръщане от една бройна система в друга, алгоритъм на Евклид, тест за прости числа с $O(\sqrt{n})$ проби за делимост, решето на Ератостен, факторизиране (с проби за делимост или решето), бързо степенуване
- ✓ Прости операции на цели числа с произволна големина (събиране, изваждане, умножение)
- ✓ Прости манипулации на масиви (попълване, преместване, ротация, обръщане, преоразмеряване, минимум/максимум, префиксни суми, хистограма, сортиране чрез броене)
- ✓ Прозорец и две показалки
- ✓ Прости алгоритми с низове (например наивно търсене за подниз)
- ✓ Последователно обработване/търсене и двоично търсене (също така и двоично търсене по отговора)

АЛ3б. По-сложни алгоритми

- ✓ Разширен алгоритъм на Евклид+~
- ✓ Бъкет сортиране, поразрядно сортиране
- ✓ Бързо сортиране и приложение за бързо намиране на k -тия най-малък елемент
- ✓ Алгоритми за сортиране със сложност $O(n \log n)$ в най-лошия случай (пирамидално сортиране, сортиране чрез сливане)
- ✓ Обхождане на наредени дървета (корен-ляво-дясно, ляво-корен-дясно, ляво-дясно-корен)
- ✓ Анализирание на аритметични изрази (например чрез превръщане в обратен полски запис)+
- ✓ Двоично повдигане~
- ✗ Троично търсене
- ✗ Минимакс евристика за игри, Алфа-бета отсичане
- ✗ 2-SAT
- ✗ Хеширане
- ✗ По-сложни алгоритми с низове като Рабин-Карп, KMP, Z-алгоритъм, Ахо-Корасик
- ✗ Сложни оптимизации на динамично като разделяй и владей, convex hull трик

АЛ3в. Алгоритми в графи

- ✓ Търсене в дълбочина и ширина
- ✓ Приложения на търсенето в дълбочина като топологично сортиране и Ойлерови цикли/пътища³
- ✓ Намиране на свързани компоненти и транзитивно затваряне
- ✓ Алгоритми за най-къс път (Дейкстра, Белман-Форд, Флойд-Уоршъл)
- ✓ Минимално покриващо дърво (алгоритми на Прим-Ярник и Крускал)
- ✓ Разширение на графи
- ✗ Лексикографски BFS, търсене по максимално съседство и техни приложения
- ✗ Двусвързаност в неориентирани графи (мостове, артикулационни точки)

³Обърнете внимание, че техниката с Ойлерово обхождане не е разрешена в АЛ4. Тук говорим за класическите алгоритми за намиране на пътища и цикли, които използват всяко ребро точно по веднъж.

- ✗ Свързаност в ориентирани графи (силно-свързани компоненти)
- ✗ Максимално двойкосъчетание в двуделен граф
- ✗ Максимален поток. Теорема за дуалността на поток/срез

АЛЗг. Геометрични алгоритми

Като правило, научния комитет има силно предпочитание към задачи, които използват целочислено смятане, за да се избегнат грешки с точността. Това може да включва представяне на някои сметнати стойности като точни обикновени дроби, но усилено използване на обикновени дроби за сметки не се очаква.

Допълнително, ако задача използва двумерни обекти, то научния комитет предпочита задачи, в които обектите са праволинейни.

- ✓ Представяне на точки, вектори, прави и отсечки
- ✓ Проверка за колинеарни точки, успоредни/перпендикулярни вектори и завои по/обратно часовниковата стрелка (например като се използва пресмятане чрез детерминанта или кръстосано (лицево) и скалярно произведение на двумерни вектори)+~
- ✓ Изчисляването на лицето на многоъгълник по координатите на върховете му+~
- ✓ Компресия на координати
- ✓ Техника на замиращата права
- ✗ Пресичане на две прави
- ✗ Проверка дали произволен многоъгълник съдържа точка
- ✗ Дуалност между точки и прави
- ✗ Пресичане на полупространства, диаграми на Вороной, триангулация на Делоне
- ✗ Изчисляване на координатите на пресичането на окръжности с прави и други окръжности
- ✗ Линейно програмиране в три или повече измерения и неговата геометрична интерпретация
- ✗ Център на масата на двуизмерен обект
- ✗ Изчисляването и представянето на композиция на геометрични трансформации, ако познаването на линейна алгебра дава предимство

АЛ4. Структури данни

- ✓ Стекове, опашки и декове
- ✓ Структура от данни пирамида
- ✓ Познаване и използване на структури данни от STL: pair, vector, stack, queue, deque, priority queue, (multi)set, (multi)map и неподредени структури
- ✓ Представяне на графи (списък на съседите, матрица на съседство, списък на ребрата)
- ✓ Представяне на непресичащи се множества: структурата Union-Find
- ✓ Статично балансирани двоични дървета за търсене. Такива са двоично индексирани дървета (известни и като дървета на Фенуик) и сегментни дървета.
- ✓ Разредени таблици за LCA, RMQ⁴ заявки
- ✓ Вграждане на структури от данни, като например редица от множества

⁴Range Minimum/Maximum Query

- ✗ Техниката lazy propagation за сегментни дървета
- ✗ Merge-sort дърво
- ✗ Персистентни структури от данни
- ✗ Самобалансиращи се двоични дървета за търсене
- ✗ Разширени двоични дървета за търсене
- ✗ Декартово дърво
- ✗ Двумерни дървовидни структури от данни (като 2D статично балансирано двоично дърво или трийп от трийпове) за 2D заявки
- ✗ Сложни варианти на пирамиди като биномна и Фибоначиева пирамида
- ✗ Trie
- ✗ Структури от данни за низове като суфиксен масив/дърво/автомат
- ✗ Декомпозиция на статични дървета (тежко-лека декомпозиция и структури от сепаратори като центроидна декомпозиция)
- ✗ Техниката с Ойлерово обхождане
- ✗ Структури от данни за динамично променящи се дървета и тяхната употреба в графови алгоритми
- ✗ Използване и имплементиране на хеш таблици (включително стратегии за разрешаване на колизии), но се очаква познаването и използването на неопределените структури на STL

АЛ5. Разпределени алгоритми – ✗

АЛ6. Криптографски алгоритми – ✗

АЛ7. Паралелни алгоритми – ✗

2.3. Други области

- Основна изчислимост – ✗
- Класовете P и NP – ✗
- Автомати и граматика – ✗