

Доклад
А група, 2024 – 2025

Атанас Димитров, Емил Инджев

21 септември 2025 г.

1 Състезания

1.1 Есенен турнир, София (47 участници)

Таблица 1: Дистрибуция по точки (Всички/Нац./Ж.Нац.)

	100	[80, 100)	[60, 80)	[40, 60)	[20, 40)	(0, 20)	0
walls	1/1/0	0/0/0	2/1/0	18/7/3	7/1/0	11/0/3	8/0/0
circle	2/2/0	2/2/0	4/1/1	2/2/0	2/0/0	2/1/0	33/2/5
xor-set	3/3/0	0/0/0	0/0/0	15/6/4	12/0/0	3/0/0	14/1/2

1.1.1 Walls

Автор: Атанас Димитров

Задачата изисква стандартния подход за преформулиране като динамично програмиране. За пълния брой точки се изисква оптимизиране на наивното динамично чрез по-сложни структури от данни като авторовото използва комбинация от дърво на Фенуик, convex-hull трика и ldl трика за допълнителна оптимизация. Единственият човек със 100 има алтернативно решение с merge-sort дърво.

1.1.2 Circle

Автор: Радослав Димитров

Геометрична задача, изискваща от състезателите да открият няколко скрити окръжности като имат единствено точки по техните очертания. Както повечето интерактивни задачи голяма част от състезателите имат 0 точки, но резултатите на хората с точки са добри. Авторовото решение използва hill-climbing като “познава” някакво начално разположение на окръжностите и се опитва да го подобри.

1.1.3 Xor Set

Автор: Иван Луков

В задачата се изисква по зададено множество S да се преброят всички числа със свойството $S = \{x \oplus k \mid x \in S\}$, където $\oplus$ е обозначено побитовото изключващо или. Авторовото решение използва вариант на бързата трансформация на Фурие (FFT) за побитовото изключващо или. За-ради доста слаби тестове участниците със 100 имат асимптотично по-лошо решение (квадратно).

1.2 НОИ 1

1.2.1 Table

Автор: Радослав Димитров

Задача за намиране на броя пътища в таблица, които имат сума на стойностите, деляща се на P . Пълното решение изисква от състезателите да използват техниката “среща-по-средата”, като намират броя пътища от началото (горе-вляво) до диагонала и от края (горе-вляво) до диагонала и комбинират за всяка клетка по този диагонал допълващите се пътища.

1.2.2 Prim

Автор: Атанас Димитров

В задачата трябва да се броят прости числа в интервал $[A, B]$ за $1 \leq A \leq B \leq 10^{12}$. Допълнително за интервала се знае че $B - A \leq 10^7$. За решението на задачата трябва да се направят няколко наблюдения за съществуването на сравнително малък ($\leq 10^6$) делител на всяко съставно число $\leq 10^{12}$ и да се модифицира стандартния алгоритъм на “решетото” на Ератостен.

1.2.3 Bipartite

Автор: Иван Лупов

Стандартна задача за определяне дали граф е двуделен. Интересно е поддържането на структура позволяваща добавянето на ребра и определяне дали новополученият граф е двуделен, което е имплементирано чрез DSU/Union-Find.

1.3 НОИ 2, 72 участници

Таблица 2: Дистрибуция по точки (Всички/Нац./Ж.Нац.)

	100	[80, 100)	[60, 80)	[40, 60)	[20, 40)	(0, 20)	0
graph	11/8/1	0/0/0	2/0/1	19/1/3	18/1/4	7/0/0	15/0/1
tree	20/10/3	0/0/0	3/0/0	4/0/1	0/0/0	17/0/5	28/0/1
kth	8/8/1	0/0/0	0/0/0	9/2/3	0/0/0	39/0/6	16/0/0

1.3.1 Graph

Автор: Радослав Димитров

Стандартна задача за намиране на най-кратък път от даден връх до всички останали върхове в граф. Интересен е начинът на описване на графът – M тройки (u, l, r) , които задават ребра от върхът u до върховете в $[l, r]$. В наивното решение би се наложило да се запишат $O(N^2)$, което дава долна граница на сложността на Дейкстра в този граф на $O(N^2)$. За пълното решение може да се експлоатира структурата на групите от заявки и чрез сегментно дърво да се забърза *PopMin* операцията в приоритетната опашка асоциирана с Дейкстра за $O(N \log^2(N))$.

1.3.2 Tree

Автор: Иван Лупов

Стандартна задача за изчисляване на обща статистика в дърво за всички двойки $[l, r]$. За решаване на задачата се изисква преобразуване на сумиране от такова, което се случва по $[l, r]$ до такова, което изчислява приноса на всеки връх в общата сума. За ефективното изчисляване на индивидуалния принос е използвано DFS-обхождане с техниката на ролбакване в структура от данни.

1.3.3 Kth

Автор: Атанас Димитров

Задача изискваща поддържане на статистика за сортирано множество от елементи след добавяне на елементи. За наивното решаване на задачата са налични много различни подходи с многообразни сложности. За пълния брой точки състезателите трябваше да съставят 2 различни решения – за малко и голямо k (спрямо корен N), и да ги комбинират в зависимост от входа.

1.4 НОИ 3 – Ден 1, Велико Търново (62 участници)

Таблица 3: Дистрибуция по точки (Всички/Нац./Ж.Нац.)

	100	[80, 100)	[60, 80)	[40, 60)	[20, 40)	(0, 20)	0
mushrooms	20/14/2	0/0/0	0/0/0	29/2/8	12/0/2	1/0/0	0/0/0
pointsort	2/2/0	0/0/0	0/0/0	14/8/1	7/2/2	19/3/4	20/1/5
candies	1/1/0	0/0/0	1/1/0	0/0/0	1/1/0	29/12/4	30/1/8

1.4.1 Mushrooms

Автори: Атанас Димитров, Енчо Мишинев

Задачата е доста лесна и стандартна – идеята ѝ е да разграничи разширения отбор и да служи за подбора за EGOI, както и да има по какво да работят по слабите участници. Трябва или гъбките, или кошничките да се разделят на тежки и леки (дали имат повече или по-малко от корен кошнички/гъбки). Така при тежките обекти, итериране само през тежките им съседи, а при леките – през всички. Ключово е, че има най-много корен тежки обекта.

1.4.2 Pointsort

Автор: Емил Инджев

Тази задача е по-ад-хок и интерактивна. Търси се да се разшири някой стандартен алгоритъм за сортиране към няколко координати. Полезно е да разгледат нужния брой сравнения за 100 точки, за ориентир колко добро решение се търси/е възможно. Има решения за 100 на база куйк сорт, както и на инсършън сорт с двоично търсене. При куйк сорт, правим разделяй и владей на 2^K части и после мърджваме по всяка координата по оптимален начин. Важно е да имаме кеш от вече правени сравнения. Помага и да избираме пивота чрез по-малка сортировка. Решението с инсършън сорт поддържа долна и горна граница за текущия елемент по всяка координата. Така избираме най-добрата точка за сравнение на база евристика (например ентропия). За да не е бавно това, трябва да разглеждаме само някои обещаващи точки (например оптималните за всяка отделна координата). Междинни точки са възможни с отделни мърдж сортове по всяка координата с общ кеш, защото кешът помага много в долните нива.

1.4.3 Candies

Автор: Атанас Димитров

Задачата е класическа по стил, но относително трудна. Тук трябва да разглеждаме свойството като мачинг и там да приложим теорема на Хол. След това е нужно да имплементираме сегментно дърво с лейзи пропъгейшън и сложен мърдж. За всеки връх пазим обща стойност, минимален подинтервал, минимален префикс и минимален суфикс (по разлика между налични и желани бонбони). Даден интервал е добър, ако минималния подинтервал в него е неотрицателен.

1.5 НОИ 3 – Ден 2, Велико Търново (61 участници)

Таблица 4: Дистрибуция по точки (Всички/Нац./Ж.Нац.)

	100	[80, 100)	[60, 80)	[40, 60)	[20, 40)	(0, 20)	0
cactus	4/4/0	0/0/0	1/1/0	1/1/0	5/2/1	34/6/8	18/2/3
stock	5/4/0	0/0/0	0/0/0	18/8/5	12/3/2	18/0/5	10/1/0
monkey	3/3/0	0/0/0	3/3/0	1/1/0	0/0/0	2/2/0	54/7/12

1.5.1 Cactus

Автори: Емил Инджев, Иван Лупов

Това е задача в стил енкоуд-декоуд. Тук ключовото е добра представа за свойствата на обратните ребра в ДФС обхождане на кактус – а именно, че всяко дървово ребро е “покрито“ от най-много едно обратно ребро. Решението първо кодира ДФС дървото. След това: или за всяко ребро маркира дали е “край“ на цикъл, или за всеки връх дали има обратно ребро и до колко над него (в единична система, като вадим едно от стойността; важно е и да пропускаме върховете които вече са покрити от децата си. Междинни решения кодират ребрата в ДФС обхождането по по неефикасни начини.

1.5.2 Stock

Автор: Радослав Димитров

Задачата е предназначена за сравнително лесна, защото се оказва, че грийди решението е вярно: всеки път или добавяме най-доброто купуване-продаване в интервал, в който нямаме обекта, или най-доброто продаване-купуване в интервал, в който имаме обекта. Поддържаеме тези интервали (и най-добрите им интервали за сделки) в приоритетна опашка. Има и алтернативно решение, което избира интервала чрез сегментно дърво. Един участник има и по-хитро решение със само един пас със стек.

1.5.3 Monkey

Автор: Михаил Банков

Задачата е леко извън ИОИ конспекта. Трябва да се разпише формулата за очакван брой ходове използвайки фейл линковете (от КМП) на низа. След това, чрез проста алгебрична манипулация можем да получим зависимост, в която елемент зависи само от предишни елементи. Така прилагаме ДП, като за да не е бавно това, е важно да държим числата като рационални дробни (по модул), за да не смятаме обратни елементи на всяка стъпка. Накрая само трябва да итерираме наобратно, за да открием отговора.

1.6 IATI – Ден 1, Шумен (114 участници)

Таблица 5: Дистрибуция по точки (Всички/Нац./Ж.Нац.)

	100	[80, 100)	[60, 80)	[40, 60)	[20, 40)	(0, 20)	0
tunnels	0/0/0	0/0/0	0/0/0	7/0/0	18/3/0	44/10/6	45/3/6
bits_and_tree	0/0/0	0/0/0	6/2/0	23/6/1	21/4/4	13/2/6	51/2/1
self_describing	22/4/0	0/0/0	6/2/1	10/3/0	0/0/0	67/7/10	9/0/1

1.6.1 Tunnels

Автор: Емил Инджев

Задачата прилича на оптимизационна, но всъщност се търси точно оптимално решение, та си е класическа, но явно доста трудна, защото изисква 3-4 отделни трудни стъпки. За първо решение, важните наблюдения са две. Първо, в даден интервал можем да правим двоично търсене за от къде е слязъл заекът (може да питаме дали е преди или след дадена клетка). Второ, цената след това зависи само от в кой интервал е слязъл заекът. Следва, че задачата може да се реши с ДП от долу нагоре, като за даден интервал на дадено ниво можем да правим ДП за подинтервали за $O(NM^3)$ (може да се оптимизира до $O(NM^2)$ със стандартни техники). За пълно решение, трябва, вместо ДП за подинтервали, да използваме грийди на база мърджване на интервали, или по-точно решение със стек, базирано на това грийди решение: Държим стек със стриктно намаляващи цени и при добавяне на нов пробваме да мърджнем на най-лявата възможна позиция. Доста международни състезатели имат ДП-то, но никой няма грийди/стек решение.

1.6.2 Bits and Tree

Автор: Виктор Кожухаров, Емил Инджев

Задачата е нещо като енкоуд-декоуд, но вместо да кодираме обект чрез битстринг, кодираме битстринг чрез обект. Освен това обекта бива модифициран, така че е нужен и error correction. Решенията се базират на това да ограничим структурата на продуцираното дърво – да има основен път като скелет и на всеки връх да се енкоудва някаква поредна част от битстринга, чрез поддърво стърчащо от пътя. Ключова е идеята да добавим хеш към битстринга – така вместо конструктивна корекция на грешката (и потенциално откриване на пътя скелет), можем да тестваме всички възможни опции и да видим при коя хешът е правилен. Това добавя около $\log(\text{Options})$ бита за кодиране, за да има голям шанс да успеем. Алтернативно решение е да се енумерират повечето (с някакво ограничение) възможни коренувани дървета използвайки дълги числа. Отново е важна идеята с хеш, за откриване на корена и на добавения връх.

1.6.3 Self-describing

Автор: Иван Лупов

Задачата дефинира специални интервали, в които всяко присъстващо число x се среща точно x пъти. На заявка се търси броя специални под-интервали на интервала $[l, r]$. Задачата е сложна дори за една заявка, та са предвидени 51 точки. Използва се “помитаща права” и наблюдения по ограниченията, идващи от присъствието на всяка различна стойност: Когато индекс i е най-десния с число $a[i]$ в даден специален подмасив, единствените валидни позиции за ляв край на подмасива са между $(a[i] + 1)$ -тото и $a[i]$ -тото срещане на числото преди индекс i . Доказуемо е, че броят на специалните интервали е малък – лесна горна граница е $O(n\sqrt{n})$, но може да се докаже и $3n$. Така втората част от задачата (обработването на заявките) се свежда до първо да открием всички специални интервали и после да отговаряме на заявки “колко от предварително открити интервали биват покрити напълно от интервал $[l, r]$?”, което е стандартна задача за персистентно дърво или merge sort дърво.

1.7 IATI – Ден 2 (Пролетно състезание), Шумен (112 участници)

Таблица 6: Дистрибуция по точки (Всички/Нац./Ж.Нац.)

	100	[80, 100)	[60, 80)	[40, 60)	[20, 40)	(0, 20)	0
soft_drinks	0/0/0	0/0/0	12/1/0	2/1/0	10/1/0	59/10/9	29/3/1
cycle	27/4/0	0/0/0	5/1/0	23/8/3	30/3/6	21/0/2	6/0/0
kitten	12/2/0	0/0/0	3/0/0	18/8/1	3/1/1	63/5/7	13/0/2

1.7.1 Soft Drinks

Автори: Атанас Димитров, Радослав Димитров

Задача със “офлайн” заявки, в която се изисква да се намери най-голямото количество, което може да се смеси от различни видове напитки при наличност на дадени ограничения за концентрацията на различните разтвори. Обективно една от най-трудните задачи през годината – изисква разглеждане на обектите в задачата като точки в координатна равнина. Нужни са сложни геометрични наблюдения за структурата на оптималния отговор и разположението му в изпъкналата обвивка на точките. За довършване на задачата за 100 точки състезателите трябваше да успеят да комбинират всички геометрични разсъждения с метода на кореновата декомпозиция за да оптимизират своето решение.

1.7.2 Cycle

Автори: Атанас Димитров, Енчо Мишинев

Задачата е доста лесна интерактивна за “сортиране”, но не типично, защото нямаме сравнение. Важно е да разберем как се държи броят цикли при размяна: или се качва с 1, ако елементите са били в един цикъл (и сега са в различни), или пада с 1, ако елементите са били в различни цикли (и сега са в един). Идеята за пълното решение е първо да сложим всички елементи в един цикъл (пробваме всеки елемент с един). След това можем да правим разделяй и владей: разменяме два произволни елемента; това най-често разделя цикъла на два приблизително еднакво големи цикъла. След това пробваме (и връщаме) да разменим всеки елемент с един, за да открием кои елементи са в кой цикъл. Повтаряме това рекурсивно, докато всички елементи не станат в различни цикли.

1.7.3 Kitten

Автор: Timur Degteari

Тази задача е сравнително лесна класическа. Важни са типичните наблюдения за диаметри в дърво – че всеки диаметър има еднакъв център (или връх или ребро). Ако е ребро, винаги бихме вадили от него. Ако е връх, бихме вадили от всички ребра на центъра, които лежат на диаметри. Т.е. грийди решението е вярно. От там има няколко възможни решение. Едно е просто да сливаме върхове, когато реброто между тях стане с тегло 0. Така (след първото централно ребро) центъра е просто един разширяващ се сет от слети върхове. За последната подзадача е важно да пазим съседите на този сет в приоритетна опашка по теглата им (така следим колко вадим от всички ребра в нея и когато това надвиши най-малкия елемент, го попваме, сливаме и пушваме неговите диаметрови съседи).

1.8 Контрола 1, Ямбол (15 участници)

Таблица 7: Дистрибуция по точки (Нац./Ж.Нац.)

	100	[80, 100)	[60, 80)	[40, 60)	[20, 40)	(0, 20)	0
treat	0/0	0/0	0/0	1/0	3/0	9/2	2/0
travel	2/0	0/0	3/0	0/0	0/0	10/2	0/0
cardgame	1/0	0/0	0/0	1/0	1/0	5/1	7/1

1.8.1 Treat

Автор: Деян Хаджи-Манич, Енчо Мишинев

Още една трудна геометрична задача. Главното разпределение по точки се случва между 0 и 45. В задачата се изисква намиране на оптимална стойност получена прилагайки една по една последователно линейни функции. За състезателите разбирането на условието и имплементирането на наивно решение беше в общия случай по-лесно от soft-drinks, но нямаше почти никакъв прогрес извън това. За пълното решение на задачата се изисква прилагането на convex-hull трика и нестандартни разсъждения от рода на “разделяй и владей” – като внимателен анализ на решението води до сложност от $O(N \log N)$.

1.8.2 Travel

Автор: Виктор Кожухаров

Сравнително стандартна постановка за намиране на минималната дължина на пътища от даден връх до всички останали върхове в дърво с допълнението, че от всеки връх освен “стандартните” ребра, излизат и портали, които могат да телепортират на разстояние $teleportStrength_i$ всеки ход. Наивното решение, което експлицитно използва разширения граф взема 13 точки, пълното решение не се различава по структура от наивното, но използва центроидна декомпозиция за да намира по-бързо “съседните” върхове в разширения граф за общо $O(N \log(N))$. Алтернативни решения може да им трябва и Фенуик дърво или да използват heavy-light декомпозиция вместо центроидна.

1.8.3 Card Game

Автор: Енчо Мишинев

Задачата е с нестандартна постановка на колаборативна игра с ограничена комуникация – играчи в кръг, всеки от които трябва да познае своята карта като вижда единствено картите на останалите играчи. Играта се развива последователно в този кръг, като играчите могат или да пропуснат хода си, или да познаят своята карта, а отделен играч им дава подсказка в началото. Решението не се базира на теоретични познания, а само на измисляне на схема. Има решения с относително “типични” подходи (надграждане на видовите случаи, с които може да се оправят играчите, чрез различни комутативни процедури), но също има и друго хитро (но по-просто) решение, което се възползва от конкретни за тази задача числови наблюдения. Задачата има и много подзадачи за частни случаи, които насочват и към двата вида решения.

1.9 Летен турнир, Пловдив (39 участници)

Таблица 8: Дистрибуция по точки (Всички/Нац./Ж.Нац.)

	100	[80, 100)	[60, 80)	[40, 60)	[20, 40)	(0, 20)	0
ops	0/0/0	0/0/0	1/1/0	0/0/0	3/3/0	22/10/5	13/2/3
subway	0/0/0	0/0/0	2/2/0	1/1/0	1/1/0	9/7/1	26/5/7
nmst	0/0/0	5/4/0	5/4/0	3/2/1	0/0/0	23/6/7	3/0/0

1.9.1 Ops

Автор: Емил Инджев

Задачата е доста сложна и нестандартна. Тук машината няма “сетива”, т.е. се търси поредица от инструкции без if-ове, която винаги работи. Задача е за сортиране, като бързо трябва да се сетим, че без if-ове, трябва да използваме sorting network. Не е нужно да правим някой особено по-сложен от бълбъл/инсършън. По-трудното е как да правим операцията “сравни и размени, ако трябва”. Сравнението е лесно, но контролираната размяна е по-сложна. Възможни начини са чрез: копиране, маскиране, събиране; или чрез трика с ксор. Базови решения достигат до 73к точки. За повече трябва или по-оптимална мрежа (достатъчно е просто да инсъртваме по 2 елемента наведнъж) или по-ефикасно сравняване и размяна: първо разширяваме числата до повече регистри (за да имаме свободни битове) и така комбинираме сравнението и част от размяната. Допълнителни предизвикателства са нуждата да се раздалечат елементите (за да има свободни регистри) и оправянето с променливото N (като се маскира суфикса с максимални стойности).

1.9.2 Subway

Автори: Деян Хаджи-Манич, Радослав Димитров, Мартин Копчев, Емил Инджев

Решението на задачата започва със стандартно построение – индуциран ненасочен двуделен граф. Набор цикли покриващ всички върхове в оригиналния граф е перфектен мачинг в новия граф (и обратното). Задачата е доста трудна за пълни точки, защото изисква да комбинираме две решения. По теорема на Хол, такъв двуделен регулярен граф винаги има перфектен мачинг. Това води до едното решение: откриваме мачинг, махаме тези ребра и повтаряме (алгоритъм на Кун се оказва много по-бърз за такива регулярни графи). Друго по-бързо решение (за D степен на двойката) е да покрием всички ребра (отново в двуделния граф) с набор от прости цикли; след това всички нечетни ребра отиват в група 1, а всички четни в група 2. Половината ребра на всеки връх са в група 1, а другата в група 2, което разделя графа на два регулярни. Така правим разделяй и владей (базов случай е когато всеки връх има едно ребро – т.е. мачинг). Така комбинираме двете решения като махаме един мачинг при нечетно D , а иначе разделяме и владеем. Последната оптимизация е да решим рекурсивно само група 1, а после да допълним група 2 до степен на двойката използвайки няколко от мачингите открити от първия дял (изтривайки ги като отговори) – така правим само $\log$ мачинга общо.

1.9.3 NMST

Автори: Христо Венев, Емил Инджев

Задачата е доста лесна. Има едно ключово наблюдение: при алгоритъм на Крускал, след всяко “ниво” (ребра с еднакви цени), състоянието на DSU-то е еднакво. Т.е. трябва да броим броя различни свързвания на всяко ниво поотделно и после да сметнем произведението на тези стойности. Броенето за едно ниво става с брутфорс, като колко точно бърз е (като сложност и потенциално константа) определя колко точки ще изкара решението. Най-добре е да правим мини DSU само за нивото (и то само за една компонента от него), като сме елиминирали дубликатни ребра, а самия брутфорс да става с undo-та и групе-ване. Последните 5 точки изискват много оптимизиране, но 95 са лесно постижими.

2 Обобщение

2.1 Теми, техники и структури

Представена е категоризация на теми, техники и структури, които се срещат в постановките или някое от предназначените пълни решения на задачите от годината.

Категория	Брой
Сегментно/Фенуик дърво	5
Интерактивна задача	5
Задача с граф	5
Задача с дърво	4
Комбинаторна задача (броене)	4
Динамично програмиране	3
Грийди	3
Разделяй и владей	3
Коренов трик (декомпозиция и други)	3
Теория на числата	3
Геометрична задача	3
Евристика	3
Двоично търсене и подобни	3
Енкоуд-декоуд задача	2
Convex hull trick	2
Disjoint set union	2
Мачинг	2
Пълно изчерпване	2
Сортиране със сравнения	2
Задача с робот/машина	1
Задача със затворници/колаборатори	1
Цикли в пермутация	1
Центроидна декомпозиция	1
Стрингове (КМП)	1
Задача за вероятност/очаквана стойност	1
Метяща права(sweep line)	1
Meet in the middle	1
Бърза трансформация на Фурие	1

2.1.1 Автори

Представяме много приблизителни бройки задачи от всеки автор, като НОИ 1 се брой с коефициент $1/2$, а НОИ 2 – с $2/3$. При няколко автора, основният автор (изготвял повечето материали) има коефициент $2/3$, а останалите разделят $1/3$. В таблицата даваме и задачите от ЕЮИ (изключвайки чуждестранните автори), тъй като работния състав е доста подобен.

Автор	А група	ЕЮИ	Общо
Атанас Димитров	5.1(6)	0	5.1(6)
Емил Инджев	4.(4)	1.(6)	6.(1)
Иван Лупов	3.5	0.(6)	4.1(6)
Радослав Димитров	3.6(1)	0	3.2(7)
Виктор Кожухаров	1.(6)	1.1(6)	2.8(3)
Енчо Мишинев	2	0	2
Деян Хаджи-Манич	1.(3)	0	1.(3)
Михаил Банков	1	0	1
Timur Degteari	1	0	1
Христо Венев	0.(6)	0	0.(6)
Мартин Копчев	0.(1)	0	0.(1)
Добрин Башев	0	0.(6)	0.(6)
Илиян Йорданов	0	0.(3)	0.(3)