

1. Разпределение на медали по точки на IOI 2016 и IOI 2017

	IOI 2106		IOI 2017	
	До място	До точки	До място	До точки
Златни	26	416	26	353
Сребърни	74	328	76	249
Бронзови	154	240	152	137

2. Разпределение на точки по задачи

IOI 2016

<u>No.</u>		Name	Max. Score	<u>Average Score</u>		Full Solutions	
Day	Task			Abs.	Rel.	<u>Abs.</u>	<u>Rel.</u>
1	1	Detecting Molecules	100	66.02	66.02%	159	50.96%
	2	Roller Coaster Railroad	100	25.53	25.53%	4	1.28%
	3	Shortcut	100	18.35	18.35%	1	0.32%
2	4	Paint by Number	100	63.6	63.60%	91	29.17%
	5	Unscrambling a Messy Bug	100	42.16	42.16%	62	19.87%
	6	Aliens	100	17.86	17.86%	1	0.32%
				38.92		53	

IOI 2017

<u>No.</u>		Name	Max. Score	<u>Average Score</u>		Full Solutions	
▲				Abs.	Rel.	<u>Abs.</u>	<u>Rel.</u>
Day	Task						
1	1	Nowruz	100	34.51	34.51%	0	0.00%
	2	Wiring	100	26.4	26.40%	45	14.42%
	3	Toy Train	100	10.43	10.43%	4	1.28%
2	4	The Big Prize	100	54.27	54.27%	27	8.65%
	5	Simurgh	100	18.4	18.40%	4	1.28%
	6	Ancient Books	100	26.21	26.21%	15	4.81%
				28.37		15.83333	

3. Съпоставка на методите за решаване на задачите

IOI 2016	IOI 2017
<p data-bbox="488 352 806 384" style="text-align: center;"><u>Detecting Molecules</u></p> <ol data-bbox="203 443 1088 660" style="list-style-type: none">1. До 46 точки – класическа задача за раницата;2. До 69 точки – оптимизиране на решението на задачата за раницата с използване на битови маски;3. До 100 точки – грийди с използване на префиксни и суфиксни суми – много познато като техника.	<p data-bbox="1498 352 1621 384" style="text-align: center;"><u>Nowruz</u></p> <ol data-bbox="1115 435 2004 1385" style="list-style-type: none">1. Необичаен тип – output only, като на състезателите се предоставят входните данни от тестовете. Оценяването е спрямо авторовото решение.2. Макар и да не е напълно „маратонска“ (задачата не е NP- пълна и има полиномиално решение, за което и авторите казват, че не е направимо на състезание), то подходът към нея трябва да бъде „маратонски“. Първото нещо при такава задача е достатъчно бързо състезателят да се откаже от търсене на „абсолютно“ решение, което би донесло 100 точки и да се съсредоточи върху решения, които ще дават достатъчно добър резултат.3. При такава задача е много важно най-напред да се анализират данните от тестовете, за да се избере стратегия, която ще доведе до колкото може повече точки. Анализът на конкретната задача показва, че освен две празни таблици, другите тестове са генерирани случайно. Това трябва да наведе на мисълта, че няма защо да се търсят различни решения за различни случаи, а трябва да се избере някакъв общ подход.

	<p>4. Общият подход е в това да се прави обхождане на граф, като се премахват върхове, заедно с прилежащите им ребра, така че накрая да остане дърво с колкото може повече листа.</p> <p>5. Какъв начин на обхождане да се избере? Тъй като искаме да се получат колкото може повече листа, то трябва да се избере BFS. Изборът на DFS води до дълги пътища от корена до листата и до малък брой листа.</p> <p>6. Трябва да се пробва от няколко начални точки за всеки тест и да се избере решението, което дава най-много точки за теста.</p>
<p style="text-align: center;"><u>Roller Coaster Railroad</u></p> <p>1. До 34 точки – динамично с битови маски – среща се в частен случай на задачата за търговския пътник.</p> <p>2. До 100 точки – конструиране на специален граф (много трудно за измисляне) и сметки върху него, включително Ойлеров цикъл намиране на минимално покриващо дърво.</p>	<p style="text-align: center;"><u>Wiring</u></p> <p>1. До 7 точки - matching или бавно динамично.</p> <p>2. Още 13 точки – директна формула.</p> <p>3. До 100 точки – доста наблюдения (базово, че има оптимално решение, в което кабелите не се пресичат). Още някои, въз основа на които динамичното става линейно. <i>Ако не се измисли хитрото динамично, то се остава на 20 точки.</i></p>

<p style="text-align: center;"><u>Shortcut</u></p> <ol style="list-style-type: none"> 1. До 23 точки – просто смятане с предварително изчисляване на определени стойности (съвсем лесно се вижда); 2. До 38 точки – пресмятане на дължините на пътища по цикъл с използване на 2 опашки (позната техника); 3. До 71 точки – смяна на подхода – двоично търсене по отговора, използване на замитаща права; Други оптимизации, които водят до 71 точки, но не дават подход нататък. 4. Следващите точки изискват специфични математически наблюдения, които градират, за да се стигне до 100 точки (най-напред се минава през използването на интервално дърво, а следващите наблюдения позволяват интервалното дърво да се замени със сортирани масиви). 	<p style="text-align: center;"><u>Toy train</u></p> <ol style="list-style-type: none"> 1. За 5 точки – печеливши и губещи позиции – елементарно. 2. Още 10 точки – пълно изчерпване. 3. Още 11 точки - намиране на цикъл – напр. DFS от всеки връх. 4. Още 11 точки – същото. 5. До 100 точки – итеративно построяване на множество от върхове, в които единият, попадайки в тях, ще спечели.
<p style="text-align: center;"><u>Paint by numbers</u></p> <ol style="list-style-type: none"> 1. До 59 точки – гриди 2. До 100 – динамично, което се измисля по-лесно от гридното. 	<p style="text-align: center;"><u>The Big Prize</u></p> <ol style="list-style-type: none"> 1. За 20 точки – стандартно двоично търсене. 2. За 70 точки – модифицирано двоично; линейно със скачане по специален начин. 3. Последните 10 точки – оптимизация на предното.

<p style="text-align: center;"><u>Unscrambling Messy Book</u></p> <p>1. Разделяй и владей. Двоични операции.</p>	<p style="text-align: center;"><u>Simurgh</u></p> <p>1. До 51 точки – една заявка за едно ребро. Използване на цикли. Намира се един цикъл, който съдържа поне едно ребро, на което не се знае вида. Пита се за всяко покриващо дърво, което не съдържа точно едно ребро от цикъла.</p> <p>2. До 100 точки - използвайки модификация на предната процедура, се намира покриващо дърво, за всяко ребро на което се знае от какъв вид е. Използвайки това дърво, могат да се правят заявки за произволно подмножество, което не съдържа цикъл. Това е ключов момент, който опростява задачата. От там се отива на двоично търсене или нещо друго.</p>
<p style="text-align: center;"><u>Aliens</u></p> <p>1. Нетривиално наблюдение, което да доведе до правилното динамично.</p> <p>2. Динамично с каквито ви дойде на ума оптимизации.</p> <ul style="list-style-type: none"> - Оптимизация на Кнут; - Разделяй и владей оптимизация; - Convex hull trick оптимизация; - Китайска оптимизация – така е разговорно между нашите състезатели. <p>Последните две оптимизации са достатъчни за 100 точки.</p>	<p style="text-align: center;"><u>Ancient books</u></p> <p>1. За 12 точки – пълно изчерпване.</p> <p>2. До 50 точки – съображение, че всички движения могат да се разбият на движения между съседни маси. След това съответният граф трябва да се допълни до ойлеров цикъл.</p> <p>3. До 100 точки – не знаая. Подозрения, че върху същия граф се прави минимално покриващо дърво, но нямаме решение.</p>

4. Обзор на задачите за А група от вътрешните състезания за 2016/2017 година

Състезание	Задача	Автор	Тематика
Есенен	Кули	Руско Шиков	Доста стандартни съображения. Персистентен стек.
	Играта „Делимост“	Даниел Атанасов	!!! Нестандартна задача с много наблюдения и несложни математически съображения.
	Биатлон	Иво Дилов	Геометрична интерпретация от линейните оптимизационни задачи. Изпъкнала обвивка.
Зимни	Хвърчила	Александър Георгиев	Обикновено динамично с използване на map и оптимизации за около 80 точки. Динамично с „разделяй и владей“ за 100 точки.
	Стена	Йордан Чапъров, Руско Шиков	Префиксни суми, индексно дърво. Lazy propagation в интервално дърво.
	СМУК	Евгений Василев, Искрен Чернев	Нестандартна. Даден е елементарен асемблероподобен език, на който трябва да се напише и изпрати към системата текст на програма, която решава конкретен проблем.

НОИ национален	Правоъгълно царство	Павел Петров	Стандартна за интервално дърво.
	100..0	Николай Белухов (реализация Павлин Пеев)	Нестандартна с доста математика. Изисква доста наблюдения и извеждане на правила.
	Намерете минимума	Антон Анастасов (с развитие от Даниел Атанасов и Йордан Чапъров)	!!! Нестандартна с много наблюдения. Без сложни структури и алгоритми.
	Триъгълник	Антон Анастасов	Доста стандартна, класическа задача за работа със структури, позволяващи бързо добавяне и премахване на елементи, както и търсене на минимален и максимален елемент. Използване на set и map от STL (може и с пирамиди).
	Monopoly	Александър Георгиев	Решение за 80 точки с техника, използваща разбиване на масив на групи от по \sqrt{N} числа и стек. Решение за 100 точки, използващо техника, подобна на двоично търсене. Има и по-лесно решение с опашка, което повечето състезатели са писали.
	Пъзел	Антон Анастасов	Динамично оптимизиране. Индексно дърво.

Пролетен	Мишки	Руско Шиков	Динамично. Разделяй и владей. Асинхронно обхождане на масиви. Доста стандартно.
	Земедел(ен)ие	Николай Белухов (релизация Павлин Пеев)	Не много сложна геометрия, но с използване на непрекъснатост и монотонност на функция и двоично търсене. Учениците не са свикнали да работят с непрекъснати функции.
	Брой правоъгълници	Даниел Атанасов	Наблюдения и оригинални обхождания на матрица. Почти същата задача я имаше на BOI 2017 (Monsters)
Контролно 1 (Видин)	Магистрали	Руско Шиков	Двоично търсене по отговора. Съображения, които водят до линейни части от алгоритъма.
	Влакчета на ужасите	Александър Георгиев	Трудна за формализиране. 2-SAT задача. Само двама човека са си представили добре какво се иска и са я решили за 100 точки. Всички други – 0.
	Изпъкнала обвивка	Руско Шиков	Техника, подобна на алгоритъма на Грахам за намиране на изпъкнала обвивка (с използване на два стека). Наблюдение, което позволява да заменим единия стек с масив и да прилагаме двоично търсене.

Контролно 2 (В. Търново)	Двоично търсене	Антон Анастасов	Сложно авторово решение. Съществува, обаче, доста просто решение, което в една или друга форма беше намерено от повечето състезатели. Конструирание на троично дърво с последващо обхождане и пресмятане по лесни за измисляне формули.
	Умножение на полиноми	Йордан Чапъров	Нестандартна!!! Съобразяване, че полиномът може да бъде представен чрез стойностите си в определен брой точки – съответно умножението на полиноми може да се представи като умножение на стойностите им в съответните точки. Интерполация.
	Consistency	Александър Георгиев	Алгоритъм на Рабин-Карп, сегментни дървета, хеширане. Изглежда трудна, но всичко е доста стандартно и почти всички са я решили.

5.Европейска младежка олимпиада по информатика

Задача	Макс. резултат	Среден резултат	Брой решения за 100 точки
magic	100	39.89	19
particles	100	25.56	13
six	100	14.74	1
camel	100	13.48	2
experience	100	17.98	7
game	100	43.43	5

Ден	Задача	Автор	Тематика
1	Six	Александър Георгиев	До 60% - възможност за решение с оптимизирано изчерпване. За повече – динамично оптимизиране със специални нестандартни наблюдения. Използване на битови маски, STL map.
	Particles	Руско Шиков, Енчо Мишинев	Лека физика – намиране на момента на сблъскване на две частици. Двоично търсене по момента на първия сблъсък. След като намерим първата двойка, сблъскали се частици, то ги премахваме и продължаваме по същия начин.
	Magic	Илиян Йорданов	Работа с низове. Интересни преформатирания на данните от символи към числа. Чрез използване на префиксни суми от решение $O(n^3)$ може да се доведе до $O(n \log(n))$.
	Experience	Радослав Димитров	Разбиване на граф-дърво с тежести на пътища. До 40-50 точки – просто динамично по дърво със сложност $O(N^2)$. За решение за 100 точки – две наблюдения: <ul style="list-style-type: none"> - Оптимално решение може да се постигне с пътища, в краищата на които се намират върховете с минимална и максимална стойност. - Оптимално решение може да се постигне с пътища, в които стойностите на върховете са в ненамаляващ или ненарастващ ред. <i>Само първото е достатъчно да се направи линейно динамично, използвайки DFS. Второто също води до такова динамично, но е по-просто за реализация.</i>

2	Camel	Павлин Пеев	<p>Нестандартна, конструктивна задача. Условието подсказва, че $n=5$ играе ключова роля. Изчерпване с връщане назад, за да се реши задачата за $n=5$. Съобразяване, че с изчерпване може да се намери хамилтонов път от всяка до всяка клетка за дъска 5×5. Разбиване на цялата дъска на квадрати 5×5 ($N=5k$). Използване на подход, в който се намира хамилтонов път в първия квадрат 5×5, който „прескача“ в съседен квадрат 5×5 и т.н. Съобразяване в каква последователност трябва да се обхождат квадратите 5×5 – има разлика за четно и нечетно N.</p>
	Game	Каспер Walentynowicz	<p>Просто съобразяване – който е на ход трябва да взема най-голямото число от множеството. Трябва структура, която да позволява добавяне на число и бързо намиране и премахване на най-голямото - пирамида или приоритетна опашка. Това решение е със сложност $O(n \log N)$ и носи 50 точки. Ключово досещане за линейно решение и 100 точки: ако поредното число, което се взема от редицата и се слага в множеството, е по-голямо от всички числа в множеството и ще бъде взето на следващия ход, то няма защо да бъде вкарвано в множеството. Тъй като числата са не по-големи от N, то можем да поддържаме масив, в чийто елемент с номер j се поддържа броя на срещанията на j в множеството. В даден момент най-голямото число, което се среща в множеството е равно на максималния индекс на елемент от масива, чийто стойност е по-голяма от 0. Трябва да имаме указател към най-десния ненулев елемент от масива. И така: ако поредното число, което се взема от редицата е по-голямо от стойността на този указател, то това</p>

		<p>число направо се добавя към резултата на играча, който е на ход, без да се прави нищо с масива; ако е по-малко или равно, то елементът с индекс, равен на числото, се увеличава с 1. За да вземе играчът, който е на ход, най-голямото число от множеството, то указателят на масива се придвижва наляво докато се срещне елемент, по-голям от 0 – неговият индекс е търсеното най-голямо число.</p> <p><i>Само 6 човека са се сетили за това решение!!!</i></p>