

# İnformatika fənni üzrə Respublika Fənn Olimpiadalarının proqramı (2024-2025-ci tədris ili)

## Giriş

Aşağıda informatika fənni üzrə Respublika Olimpiadalarının tam proqramı verilir. Bu proqram **1. Riyaziyyat, 2. Kompüter elmi, 3. Proqram mühəndisliyi və 4. Kompüter savadlılığı** başlıqları altında əsas 4 bölmədən ibarətdir. Hər bir bölmə daxilində ən alt mövzu və anlayışların hansı yaş (sınıf) kateqoriyasına aid olduğunu göstərən [] daxilində ədəd verilir. Burada, [8] 8-ci sinif və daha yuxarı, [9] 9-cu sinif və daha yuxarı, [10] 10-cu sinif və daha yuxarı, [11] isə yalnız 11-ci sinifə aid olan mövzu və anlayışları bildirir.

Qeyd edək ki, Respublika mərhələsində (yarımfinal və final turları) aşağı yaş qrupu (8-9-cu siniflər) üzrə [8] və [9] ilə işarələnmiş mövzular, yuxarı yaş qrupu (10-11-ci siniflər) üzrə isə bütün mövzular olimpiada proqramına daxildir.

## 1. Riyaziyyat

### 1.1. Hesab və Həndəsə

- 1.1.1. Tam ədədlər, onlar üzərində əməliyyatlar və onların müqayisəsi [8]
- 1.1.2. Tam ədədlərin əsas xassələri (işarə, paritet, bölünmə) [8]
- 1.1.3. Sadə modul hesablamalar: toplama, çıxma, vurma [8]
- 1.1.4. Sadə ədədlər, sadə vuruqlar, ƏBOB, ƏKOB anlayışları [8]
- 1.1.5. Kəslər, faizlər [8]
- 1.1.6. Say sistemləri [8]
- 1.1.7. Xətti Diofant tənliyi [9]
- 1.1.8. Xətt, xətt seqmenti, bucaq, üçbucaq, düzbucaqlı, kvadrat, dairə [8]
- 1.1.9. Müstəvi üzərində nöqtə, vektor, koordinatlar [8]
- 1.1.10. Çoxbucaqlı (təpə, yan/kənar, sadə, qabarıq, daxili, sahə anlayışları) [8]
- 1.1.11. Evklid məsafəsi [8]
- 1.1.12. Pifaqor teoremi [8]

### 1.2. Diskret strukturlar

#### 1.2.1. Funksiyalar, əlaqələr və çoxluqlar

- 1.2.1.1. Funksiyalar (suryeksiyalar, inyeksiyalar, tərslər, kompozisiya anlayışları) [9]
- 1.2.1.2. Əlaqələr (refleksivlik, simmetriya, tranzitivlik, ekvivalentlik münasibətləri, leksikoqrafik sıra anlayışları) [9]
- 1.2.1.3. Çoxluqlar (daxil etmə/çıxma, tamamlayıcılar, çarpaz vurma anlayışları) [9]
- 1.2.1.4. Kardinallıq və sayma (sonlu çoxluqlarda) [9]

#### 1.2.2. Təməl məntiq

- 1.2.2.1. Birinci dərəcəli məntiq [8]
- 1.2.2.2. Məntiqi bağlayıcılar (onların əsas xassələri) [8]
- 1.2.2.3. Doğruluq cədvəli [8]
- 1.2.2.4. Universal və mövcudluq kəmiyyət göstəriciləri [8]

- 1.2.2.5. “Modus ponens” və “modus tollens” [8]
- 1.2.3. İsbat üsulları
  - 1.2.3.1. Birbaşa isbat [8]
  - 1.2.3.2. Əks nümunə, ziddiyyət göstərməklə, əksini fərz etməklə olan isbatlar [8]
  - 1.2.3.3. Qurma ilə isbat [8]
  - 1.2.3.4. Riyazi induksiya [9]
  - 1.2.3.5. Güclü induksiya (tam induksiya da adlanır) [9]
  - 1.2.3.6. Rekursiv riyazi yazılışlar [9]
- 1.2.4. Saymanın əsasları
  - 1.2.4.1. Toplama və vurma prinsipləri [8]
  - 1.2.4.2. Ədədi və həndəsi silsilə [9]
  - 1.2.4.3. Fibonacci ədədləri [9]
  - 1.2.4.4. Permutasiyalar və kombinasiyalar (əsas anlayışlar) [9]
  - 1.2.4.5. Faktorial anlayışı və binomial əmsallar [9]
  - 1.2.4.6. Daxil etmə/çıxma prinsipi [9]
  - 1.2.4.7. Dirixle prinsipi [8]
  - 1.2.4.8. Paskal qaydası, binomun açılışı [10]
  - 1.2.4.9. Catalan ədədləri [10]
- 1.2.5. Qraflar və ağaclar
  - 1.2.5.1. İstiqamətlənməmiş qraflar (təpə/düyün, til, dərəcə, qonşuluq, təpə və til etiketləri anlayışları) [9]
  - 1.2.5.2. İstiqamətlənmiş qraflar (daxili-dərəcə, xarici-dərəcə anlayışları) [9]
  - 1.2.5.3. Multiqraflar, döngülər (özünə til) [9]
  - 1.2.5.4. Qraflarda yollar (istiqamətlənməmiş və istiqamətlənmiş yol, dövr, Eylər turu, Hamilton yolu/dövrü anlayışları) [9]
  - 1.2.5.5. Birləşik komponentlər, alt-qraf və ən qısa məsafə anlayışları [9]
  - 1.2.5.6. Ağaclar (yarpaq, diametr, mərkəz, ağırlıq mərkəzi, meşə anlayışları) [9]
  - 1.2.5.7. Köklü ağaclar (kök, valideyn, uşaq, əcdad, alt-ağac anlayışları) [9]
  - 1.2.5.8. Genişlənən ağaclar (Spanning trees) [10]
  - 1.2.5.9. Axtarış strategiyaları [10]
  - 1.2.5.10. Bipartit qraflar [10]
  - 1.2.5.11. İstiqamətlənmiş asiklik qraflar [10]
  - 1.2.5.12. Planar qraflar [11]
  - 1.2.5.13. Qrafların əsas kombinator xüsusiyyətləri [11]

## 2. Kompüter elmi

- 2.1. Kompüter elmi haqqında ilkin anlayışlar
  - 2.1.1. Kompüterlərin iş prinsipi [8]
  - 2.1.2. İkilik say sistemi [8]
  - 2.1.3. ASCII kodları [8]
- 2.2. Proqramlaşdırmanın əsasları
  - 2.2.1. Proqramlaşdırma haqqında ilkin anlayışlar

- 2.2.1.1. C++ və ya pythonda proqramlaşdırma bacarığı və bu dillərdən ən az birinin sintaks və semantikasına bələd olmaq [8]
- 2.2.1.2. Verilənlər, verilənlərin tipləri, verilənlərin saxlanması, ifadələr [8]
- 2.2.1.3. Sadə giriş/çıxış funksiyaları [8]
- 2.2.1.4. Operatorlar [8]
- 2.2.1.5. Şərti və iterativ idarəetmə strukturları (şərt ifadələri, dövrlər) [8]
- 2.2.1.6. Funksiyalar və arqument ötürmə [8]
- 2.2.2. Alqoritmlər və problem həll etmə
  - 2.2.2.1. Problemi həll etmə strategiyaları [8]
  - 2.2.2.2. Problemi həll etmədə alqoritmlərin rolu [8]
  - 2.2.2.3. Alqoritmləri icra etmə strategiyaları [8]
  - 2.2.2.4. Sazlama (xətaların aradan qaldırılması) strategiyaları [8]
  - 2.2.2.5. Həll üsulunun (alqoritmin) doğruluğu, effektivliyi anlayışları [8]
- 2.2.3. Fundamental verilənlər strukturları
  - 2.2.3.1. Primitiv tiplər (boolean, signed/unsigned integer, character) [8]
  - 2.2.3.2. Massivlər (çox ölçülü massivlər də daxil) [8]
  - 2.2.3.3. Sətirlər (strings) və sətirlərin emalı [8]
  - 2.2.3.4. Statik və stek yaddaşı (elementar avtomatik yaddaş idarəetməsi) [9]
  - 2.2.3.5. Əlaqəli (linked) strukturlar [9]
  - 2.2.3.6. Qraflar və ağacların proqramda saxlanılma üsulları [9]
  - 2.2.3.7. Uyğun verilənlər strukturunu seçmək üçün strategiyalar [8]
  - 2.2.3.8. Həqiqi ədədlərin istifadəsi, həqiqi ədədlərin üzən nöqtəli təsviri və dəqiqliklə bağlı problemlərin mövcudluğu [9]
  - 2.2.3.9. Göstəricilər və istinadlar [9]
  - 2.2.3.10. Yaddaşda məlumatların saxlanılma şəkli [9]
  - 2.2.3.11. "Heap" yaddaşı [9]
- 2.2.4. Rekursiya
  - 2.2.4.1. Rekursiya anlayışı [9]
  - 2.2.4.2. Rekursiv riyazi funksiyalar [9]
  - 2.2.4.3. Sadə rekursiv prosedurlar (qaşılıqlı rekursiya da daxil) [9]
  - 2.2.4.4. Böl və idarə et strategiyası [9]
  - 2.2.4.5. Rekursiyanın icrası [9]
  - 2.2.4.6. Rekursiv geri izləmə (backtracking) [10]
- 2.2.5. İnteraktiv proqramlaşdırma [10]

Bəzi tapşırıqlar reaktiv mühitlə qarşılıqlı dialoq tələb edə bilər. Təmin edilmiş mühitlə belə qarşılıqlı dialoqun həyata keçirilməsi

## 2.3. Alqoritmlər və mürəkkəblilik

### 2.3.1. Əsas alqoritmik təhlillər

- 2.3.1.1. Yuxarı sərhəd mürəkkəbliyinin asimptotik təhlili [9]
- 2.3.1.2. Böyük  $O$  notasiyası [9]
- 2.3.1.3. Standart mürəkkəblilik sinifləri: sabit, loqarifmik, xətti,  $O(n \log n)$ , kvadratik, kubik, eksponensial və s. [9]
- 2.3.1.4. Zaman və yaddaşdan effektiv istifadə [9]

- 2.3.1.5. Alqoritmlərdə zaman və yaddaş uzlaşması [9]
- 2.3.1.6. Ən yaxşı, orta və ən pis performans anlayışları [10]
- 2.3.1.7. Balaca  $o$ ,  $\Omega$  (omeqa) və  $\Theta$  (teta) notasiyaları [10]
- 2.3.1.8. "Amortized" təhlil [10]
- 2.3.2. Alqoritmik strategiyalar
  - 2.3.2.1. Sadə döngü dizayn strategiyaları [8]
  - 2.3.2.2. Hərtərəfli axtarış alqoritmləri [8]
  - 2.3.2.3. Açıq alqoritmlər [8]
  - 2.3.2.4. Böl və idarə et [9]
  - 2.3.2.5. Geri izləmə (rekursiv və rekursiv olmayan) [10]
  - 2.3.2.6. Dinamik proqramlaşdırma [9]
  - 2.3.2.7. Evristika [11]
- 2.3.3. Alqoritmlər və problem həll etmə üsulları
  - 2.3.3.1. Tam ədədləri ehtiva edən sadə alqoritmlər
    - 2.3.3.1.1. Say sistemi çevrilməsi [8]
    - 2.3.3.1.2. Evklid alqoritmi [8]
    - 2.3.3.1.3.  $O(\sqrt{n})$  yoxlama ilə sadəlik testi [8]
    - 2.3.3.1.4. Sadə ədədlərin tapılması – Eratosfen xəlbiri [8]
    - 2.3.3.1.5. Sadə vuruqlara ayırma [8]
    - 2.3.3.1.6. Ədədin bölənlərinin tapılması [8]
    - 2.3.3.1.7. Effektiv qüvvətə yüksəltmə [8]
  - 2.3.3.2. Massiv üzərində sadə manipulyasiyalar (massivin doldurulması, elementlərinin sürüldürülməsi, döndərilməsi, tərs çevrilməsi, minimum/maksimum elementin tapılması, prefiks cəmi, histoqram, blok sıralaması və s.) [8]
  - 2.3.3.3. Sadə sətir alqoritmləri (məs. bəsit alt-sətir axtarışı) [9]
  - 2.3.3.4. Xəttilik axtarış və ikili axtarış [8]
  - 2.3.3.5.  $O(n^2)$  sıralama alqoritmləri (bubble sort, selection sort və insertion sort) [8]
  - 2.3.3.6. Sürətli sıralama (quicksort) və sürətli seçmə (quickselect) [9]
  - 2.3.3.7. Ən pis performans  $O(n \log n)$  olan sıralama alqoritmləri (heap sort və merge sort) [9]
  - 2.3.3.8. Sıralı ağaclarda axtarış üsulları (pre, in, post) [9]
  - 2.3.3.9. Dərininə və eninə axtarış – DFS, BFS alqoritmləri [9]
  - 2.3.3.10. DFS və BFS'in tətbiqləri
    - 2.3.3.10.1. Birləşmiş komponentlərin tapılması [9]
    - 2.3.3.10.2. Qraflarda dövrlərin tapılması [9]
    - 2.3.3.10.3. İstiqamətlənmiş qraflarda güclü birləşmiş komponentlərin tapılması [9]
    - 2.3.3.10.4. Qrafın iki rəngdə rənglənməsi / bipartit qraf [9]
    - 2.3.3.10.5. Topoloji sıralama [9]
    - 2.3.3.10.6. Qrafda körpülərin tapılması [11]
    - 2.3.3.10.7. Qrafda birləşmə nöqtələrinin tapılması [11]
  - 2.3.3.11. Ən qısa yolun tapılması alqoritmləri
    - 2.3.3.11.1. Dijkstra alqoritmi [9]
    - 2.3.3.11.2. Floyd-Warshall alqoritmi [9]
    - 2.3.3.11.3. Bellman-Ford alqoritmi [9]
    - 2.3.3.11.4. Johnson alqoritmi [11]

- 2.3.3.11.5. SPFA alqoritmi [11]
- 2.3.3.12. Minimum çəkili ağac
  - 2.3.3.12.1. Prim alqoritmi [10]
  - 2.3.3.12.2. Kruskal alqoritmi [10]
  - 2.3.3.12.3. İkinci ən yaxşı minimum çəkili ağacın tapılması [11]
- 2.3.3.13. Ən yaxın ortaq əcdadın tapılması alqoritmi – “Binary lifting” üsulu [9]
- 2.3.3.14. Ağacda Eylər turu, Eylər turundan istifadə etməklə ən yaxın ortaq əcdadın tapılması [10]
- 2.3.3.15. Maksimum ikitərəfli uyğunluğun (bipartite matching) hesablanması üçün  $O(VE)$  alqoritm [10]
- 2.3.3.16. Kombinatorial oyun nəzəriyyəsinin əsasları, uduş və uduzma mövqeləri, optimal oyun oynamaq üçün minimaks alqoritmi [10]
- 2.3.3.17. “Two pointers” üsulu [9]
- 2.3.3.18. “Sliding window” üsulu [9]
- 2.3.3.19. “Meet in the middle” üsulu [9]
- 2.3.3.20. “Sqrt decomposition” üsulu və Mo alqoritmi [10]
- 2.3.3.21. Həndəsi alqoritmlər
  - 2.3.3.21.1. Nöqtələri, vektorları, xətləri, xətt segmentlərini ifadə etmək [8]
  - 2.3.3.21.2. Kollinear nöqtələrin, paralel/ortoqonal vektorların və saat əqrəbi istiqamətində dönmələrin yoxlanılması (məsələn, skalyar və vektorial hasilərdən istifadə etməklə) [9]
  - 2.3.3.21.3. İki xəttin kəsişməsi [8]
  - 2.3.3.21.4. Təpə koordinatlarından çoxbucaqlının sahəsinin hesablanması [9]
  - 2.3.3.21.5. Nöqtənin (ixtiyari/qabarıq) çoxbucaqlının daxilində olub-olmadığını yoxlamaq [10]
  - 2.3.3.21.6. Koordinat sıxılması [9]
  - 2.3.3.21.7. Qabarıq çoxbucaqlının qurulması üçün  $O(n \log n)$  alqoritmlər [10]
  - 2.3.3.21.8. Verilmiş  $n$  nöqtə arasında ən yaxın iki nöqtənin tapılması [10]
  - 2.3.3.21.9. Süpürmə xətti (sweeping line) alqoritmi [10]
- 2.4. Verilənlər strukturları
  - 2.4.1. “Stack” və “queue” strukturları [8]
  - 2.4.2. Qrafların proqram yaddaşında saxlanma üsulları
    - 2.4.2.1. Qonşuluq matrisi [9]
    - 2.4.2.2. Qonşuluq siyahısı [9]
  - 2.4.3. Binar “heap” verilənlər strukturu [9]
  - 2.4.4. “Sparse table” strukturu [9]
  - 2.4.5. “Disjoint set union” (DSU) verilənlər strukturu [9]
  - 2.4.6. Statik balanslı ikili axtarış ağacları
    - 2.4.6.1. “Segment tree” [9]
    - 2.4.6.2. İkili indeksli ağac (“Fenwick tree”) [9]
  - 2.4.7. Balanslı ikili axtarış ağacları [9]
  - 2.4.8. Statik ağacların parçalanması
    - 2.4.8.1. “Heavy-light decomposition” [11]
    - 2.4.8.2. “Centroid decomposition” [11]
  - 2.4.9. Yeniləmə zamanı əvvəlki versiyaların saxlanması - “Persistent” anlayışı [11]
  - 2.4.10. İç-içə verilənlər strukturları (2D verilənlər strukturları) [11]
  - 2.4.11. “Trie” verilənlər strukturu [10]

### 3 Proqram mühəndisliyi

#### 3.1. Proqram dizaynı

- 3.1.1. Əsas dizayn anlayışları və prinsipləri [8]
- 3.1.2. Dizayn nümunələri [8]
- 3.1.3. Strukturlaşdırılmış dizayn [8]

Müsabiqə iştirakçılarında xüsusilə mücərrəd alqoritmi standart və ya müsabiqəyə xas kitabxanalardan istifadə etməklə icazə verilən proqramlaşdırma dillərindən birində ifadə edilmiş konkret, səmərəli proqrama çevirmək gözlənilə bilər.

#### 3.2. Tətbiqi proqramlaşdırma interfeyslərindən (API-lardan) istifadə [9]

Müsabiqə iştirakçılarında xüsusilə müsabiqəyə xas kitabxanaları təqdim edilmiş spesifikasiyalara uyğun istifadə edə bilmələri gözlənilə bilər.

#### 3.3. Proqram alətləri və mühitləri

- 3.3.1. Kod editor və ya IDE-lərdən istifadə etmə bacarığı [8]
- 3.3.2. Yazdıqları proqramları kompayl etmə və işlə salma [8]
- 3.3.3. Proqramlarında xətalara aşkarlama və düzəltmə (debugging) [8]

#### 3.4. Proqramın test edilməsi

- 3.4.1. Test etmənin əsasları: Test planının yaradılması, test nümunələrinin yaradılması və s. [8]
- 3.4.2. Qara-qutu və ağ-qutu test üsulları [10]
- 3.4.3. Vahid, inteqrasiya, doğrulama və sistem testi [10]

Müsabiqə iştirakçılarında xüsusilə tez-tez rast gəlinən səhvləri aşkar etmək imkanını maksimum dərəcədə artıran üsulları tətbiq etmələri gözlənilə bilər (məsələn, yaxşı strukturlaşdırılmış kod, kodun nəzərdən keçirilməsi, test icra etmək, proqramın ayrı-ayrı hissələrinin test edilməsi və s. üsullarla)

#### 3.5. Proqram layihəsinin idarə edilməsi

- 3.5.1. Layihənin planlaşdırılması (xüsusilə vaxtın idarə edilməsi) [8]
- 3.5.2. Risk təhlili [8]
- 3.5.3. Proqram təminatının konfigurasiyasının idarə edilməsi [8]

Müsabiqə iştirakçılarında xüsusilə aşağıdakılar gözlənilə bilər:

- Müxtəlif fəaliyyətlərə sərf olunan vaxtı idarə etmək
- Alternativ yanaşmalar arasında seçim edərkən riskləri ölçmək
- Həllər hazırlayarkən müxtəlif versiyaları və onların vəziyyətini nəzərdə saxlamaq

## 4 Kompüter savadlılığı [8]

Müsabiqə iştirakçıları kompüterin əsas strukturunu və işini (prosessor, yaddaş, I/O) bilməli və başa düşməlidirlər. Onların qrafik istifadəçi interfeysi olan standart kompüterdən, onun dəstəkləyici proqramları olan əməliyyat sistemindən və müsabiqə tapşırıqlarının həlli üçün təqdim olunmuş proqram tərtibat alətlərindən istifadə edə bilmələri gözlənilir. Xüsusilə, faylların idarə edilməsində bəzi bacarıqlar faydalıdır (qovluqların yaradılması, faylların sürətinin çıxarılması və köçürülməsi).

Müsabiqə iştirakçılarına verilən kompüterlərdə bir sıra ekvivalent alətlər təqdim oluna bilər. Müsabiqə iştirakçılarının bütün bu alətlərin bütün xüsusiyyətlərini bilmələri gözlənilmir. Onlar ən uyğun gördükləri alətlərə əsasən öz seçimlərini edə bilərlər.

### Ədəbiyyat siyahısı

1. Azərbaycan Respublikası Təhsil Nazirliyi tərəfindən nəşr olunan Ümumtəhsil məktəblərinin informatika fənni üzrə dərs kitabları.
2. Introduction to Algorithms (Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest).
3. The Algorithm Design Manual (Steven S Skiena).
4. Antti Laaksonen-Guide to competitive programming\_learning and improving algorithms through contests-Springer International Publishing (2017).
5. Competitive Programmer's Handbook (Antti Laaksonen, July 3, 2018)
6. [Pearson] - Algorithms, 4th ed. - [Sedgewick, Wayne].
7. Competitive Programming 4: The Lower Bound of Programming Contests in the 2020s (Steven Halim, Felix Halim, Suhendry Effendy).
8. The Art of Computer Programming. (Donald E. Knuth.Addison-Wesley)
9. Alexander-Shen Algorithms and Programming.
10. The International Olympiad in Informatics Syllabus Version 2023

### Onlayn resurslar:

- <https://ioinformatics.org/files/ioi-syllabus-2023.pdf>
- <https://www.hackerrank.com/domains/cpp>
- <https://cplusplus.com/reference/stl/>
- <https://www.hackerrank.com/domains/python>
- <https://cses.fi/book/book.pdf>
- <https://cses.fi/problemset/>
- <https://usaco.guide/>
- <https://cp-algorithms.com/>
- <https://e-maxx.ru/algo/>
- <https://codeforces.com/>
- <https://atcoder.jp/>
- <https://www.eolymp.com/>

## Qeyd və tövsiyələr

Müsabiqədə iştirak edəcək şagirdlərə, olimpiadaya hazırlaşarkən öncə C++ və ya python dillərindən birinin əsaslarını daha sonra isə digər mövzuları öyrənmək tövsiyə olunur. Bunun üçün yuxarıda qeyd olunmuş ədəbiyyatlardan və onlayn resurslardan istifadə edə bilərsiniz. Həmçinin qeyd edək ki, hazırda bir çox olimpiadalarda o cümlədən beynəlxalq informatika olimpiadasında yalnız C++ dilindən istifadə etmək mümkündür və C++ dili olimpiadalarda ən geniş istifadə olunan dildir. Bu səbəbdən C++ dilini öyrənməyiniz tövsiyə olunandır.

Dilin əsaslarını öyrəndikdən sonra onlayn resurlar bölməsində qeyd olunmuş müxtəlif platformalarda sadədən mürəkkəbə tapşırıqlar həll edə və onlayn yarışlara qatıla bilərsiniz. Bunu şiddətlə tövsiyə edirik.