# LISTA DE CONTEÚDOS PARA A OLIMPÍADA BRASILEIRA DE INFORMÁTICA

## 1. Versão

Esta é a lista de conteúdos oficial para a OBI 2016. Para feedback ou sugestões entre em contato com a organização da olimpíada (olimpinf@ic.unicamp.br ou corretor.obi@gmail.com). Ela foi elaborada por Arthur Pratti Dadalto com base no *syllabus* da IOI (https://people.ksp.sk/~misof/ioi-syllabus/ioi-syllabus.pdf), que também pode ser usado como referência.

## 2. Introdução

Esta lista contém tópicos que podem ser cobrados na OBI em cada um de seus níveis seguindo a seguinte simbologia:

- [J] Pode ser cobrado a partir do Nível Junior;
- [N1] Pode ser cobrado a partir do Nível 1;
- [N2] Pode ser cobrado a partir do Nível 2 (no nível universitário podem estar presentes os mesmos tópicos do nível 2);
- [S] Pode ser cobrado na Seletiva para a IOI.

## 3. Conceitos básicos de Aritmética e Geometria

a. [J] Inteiros, operações e comparações;
b. [J] Propriedades básicas dos inteiros (sinal, paridade, divisibilidade, etc);
c. [J] Frações;
d. [J] Linha, segmento de linha, ângulo, triângulo, retângulo, quadrado, circunferência;
e. [J] Distância Euclidiana; [5]
f. [J] Teorema de Pitágoras;
g. [J] Números primos;
h. [N1] Ponto, vetor, coordenadas no plano;
i. [N2] Aritimética modular básica: adição, subtração e multiplicação; [21] [30]
j. [N2] Polígono (vértice, aresta, convexo, área);
k. [N2] Operacões com matrizes (adição, multiplicação e exponenciação).

## 4. Conceitos básicos de Matemática Discreta

a. [J] Indução matemática; [6]
b. [J] Relações (reflexão, simetria, ordem lexicográfica, etc); [7]
c. [J] Funções (injeção, inversa, composição, etc); [8]

d. [J] Conjuntos (inclusão/exclusão, complementos, produto Cartesiano, etc). [9]
e. [J] Definições matemáticas recursivas; [10]
f. [J] Princípio das casas dos pombos; [11] [18]
g. [N1] Contagem (regras da soma e do produto, progressão aritmética e geométrica, números de Fibonacci, etc); [18] [12] [13] [14] [30]
h. [N1] Definições básicas de permutação e combinação; [15] [16]
i. [N1] Função fatorial; [17]
j. [N1] Princípio da inclusão-exclusão; [18] [19]
k. [N2] Coeficientes binomiais; [20]
l. [N2] Triângulo de Pascal. [20]

**5. Conceitos de grafos e árvores**

a. [J] Arvóres e suas propriedades básicas, árvore enraizadas; [22] [23][38]
b. [J] Grafos direcionados e não direcionados; [22] [23] [38]
c. [J] Grau, caminho, ciclo, conectividade; [22] [23]
d. [N1] Grafos com pesos, cores ou classificações nas arestas ou vértices; [22] [23]
e. [N1] Grafos bipartidos; [22] [23]
f. [N1] Grafos com arestas múltiplas; [22] [23]
g. [N2] Caminho/ciclo de Euler/Hamilton. [22] [23]

**6. Estratégias de algoritmos**

a. [J] Estratégias com loop simples;
b. [J] Força bruta;
c. [J] Algoritmos gulosos; [24]
d. [J] Divisão e conquista; [25]
e. [J] Backtracking; [26]
f. [N1] Programação dinâmica. [27] [28] [29]

**7. Algoritmos**

a. [J] Algoritmo de Euclides; [30] [31]
b. [J] Teste de primalidade em $O(\sqrt{N})$; [30]
c. [J] Exponeciacão eficiente; [30]
d. [J] Arrays: máximo, mediana, soma de prefixos, histograma, etc; [32] [33]
e. [J] Algoritmos de ordenação em $O(N^2)$; [34] [35]
f. [J] Busca linear e busca binária; [36]
g. [N1] Crivo de Eratóstenes; [30] [39]
h. [N1] Teoria de jogos, posições vencedoras e perdedoras, algoritmo minimax para jogo de forma ótima; [40]

i. [N1] Algoritmos de ordenação em $O(NlogN)$: heap sort, merge sort, etc; [34] [35]
j. [N2] Operações simples em inteiros de tamanho arbitrário; [41]
k. [N2] Algoritmos de força bruta e programção dinâmica com auxílio de máscaras de bits; [51] [52] [53]
l. [N2] Exponenciação de matrizes para resolver problemas de programação dinâmica; [54]
m. [S] Quickselect para achar o k-ésimo menor elemento; [55]

**8. Algoritmos em grafos**

a. [J] Percorrer grafos com busca em largura e busca em profundidade; [38] [37]
b. [N1] Algoritmos de caminho mínimo (Dijkstra, Bellman-Ford, Floyd-Warshall); [46] [47]
c. [N1] Encontrar componentes conexas;
d. [N1] Ordenação topológica; [42] [43]
e. [N1] Árvores geradoras mínimas; [48] [49]
f. [N2] Encontrar um caminho/ciclo de Euler; [50]
g. [S] Conjunto de arestas independes em grafo bipartido (bipartite matching) em $O(VE)$. [56]

**9. Estruturas de dados**

a. [J] Não é necessário, mas é muito útil conhecer a STL usando C++; [59]
b. [J] Pilhas e filas; [60]
c. [J] Listas ligadas; [61]
d. [J] Representação de grafos;
e. [J] Árvore de busca binária estática; [62]
f. [N1] Heap binário; [63]
g. [N1] Conjuntos disjuntos: Union-find; [64]
h. [N1] Árvore de Fenwick (binary indexed tree) 1D; [65]
i. [N1] Menor ancestral comum: algoritmo para responder perguntas em $O(logN)$. [66]
j. [N2] Árvore de Fenwick (binary indexed tree) 2D; [65]
k. [N2] Árvore de segmentos (Segment tree); [67] [68] [69]
l. [S] Estruturas de dados persistentes;
m. [S] Divisão em buckets de tamanho $\sqrt{N}$ (square root decomposition); [1]
n. [S] Árvores de busca binária balanceadas (Treaps, splay trees, etc); [70]
o. [S] Árvore de segmentos 2D; [69]
p. [S] Tries. [71]

10. **Geometria computacional**
    a. [N1] Pontos, vetores, linhas e segmentos de linhas; [72]
    b. [N1] Pontos colineares, vetores paralelos e ortgonais. [72]
    c. [N1] Interseção de duas linhas; [72]
    d. [N2] Compressão de coordenadas; [73]
    e. [N2] Convex hull em $O(NlogN)$; [72]
    f. [N2] Line sweep; [4]
    g. [S] Calcular área de um polígono; [2] [72]
    h. [S] Checar se um polígono contém um ponto; [3] [72]

11. **Referências**


[1] http://www.infoarena.ro/blog/square-root-trick
[2] https://en.wikipedia.org/wiki/Shoelace_formula
[3] https://en.wikipedia.org/wiki/Point_in_polygon
[4]https://www.topcoder.com/community/data-science/data-science-tutorials/line-sweep-algorithms/
[5] https://pt.wikipedia.org/wiki/Dist%C3%A2ncia_euclidiana
[6] http://www.obmep.org.br/docs/apostila4.pdf
[7] http://www.ime.usp.br/~pf/algoritmos/aulas/string.html
[8] https://pt.wikipedia.org/wiki/Fun%C3%A7%C3%A3o_(matem%C3%A1tica)
[9] https://pt.wikipedia.org/wiki/Conjunto
[10] https://pt.wikipedia.org/wiki/Defini%C3%A7%C3%A3o_recursiva
[11] http://clubes.obmep.org.br/blog/texto_002-principio-das-casas-dos-pombos/
[12]http://clubes.obmep.org.br/blog/texto_006-principio-fundamental-de-contagem/principio-fundamental-de-contagem-generalizacao/
[13]http://vestibular.uol.com.br/ultnot/resumos/progressao-aritmetica-geometrica.jhtm
[14] http://www.mat.uc.pt/~mat1131/Fibonacci.html
[15] https://pt.wikipedia.org/wiki/Permuta%C3%A7%C3%A3o
[16] https://pt.wikipedia.org/wiki/Combina%C3%A7%C3%A3o_(matem%C3%A1tica)
[17] http://www.infoescola.com/matematica/fatorial/
[18] http://www.cin.ufpe.br/~gdcc/matdis/aulas/contagem.pdf
[19]https://www.artofproblemsolving.com/wiki/index.php?title=Principle_of_Inclusion-Exclusion
[20] http://www.cin.ufpe.br/~gdcc/matdis/aulas/binomial
[21]https://pt.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/modular-exponentiation
[22] https://pt.wikipedia.org/wiki/Teoria_dos_grafos
[23] http://www.ime.usp.br/~pf/teoriadosgrafos/texto/TeoriaDosGrafos.pdf
[24] http://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/guloso.html
[25]http://www.decom.ufop.br/toffolo/site_media/uploads/2011-1/bcc402/slides/08._divisao_e_conquista.pdf

[26] https://en.wikibooks.org/wiki/Algorithms/Backtracking
[27]https://www.topcoder.com/community/data-science/data-science-tutorials/dynamic-programming-from-novice-to-advanced/
[28] https://www.codechef.com/wiki/tutorial-dynamic-programming
[29]https://www.quora.com/I-want-to-learn-memoization-What-are-some-links-with-problems-from-SPOJ-Topcoder-Codeforces
[30]https://docs.google.com/presentation/d/1ABSFgyRu1I-yKyOxA6RyUqUxqmvxF7XaHySaGmUgSvc/edit?usp=sharing
[31] https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides
[32] http://codeforces.com/blog/entry/15729
[34] http://www.tutorialspoint.com/data_structures_algorithms/sorting_algorithms.htm
[35] https://pt.wikipedia.org/wiki/Merge_sort
[36] http://www.ime.usp.br/~pf/algoritmos/aulas/bubi.html
[37]https://www.topcoder.com/community/data-science/data-science-tutorials/introduction-to-graphs-and-their-data-structures-section-2/
[38]https://www.topcoder.com/community/data-science/data-science-tutorials/introduction-to-graphs-and-their-data-structures-section-1/
[39] https://pt.wikipedia.org/wiki/Crivo_de_Erat%C3%B3stenes
[40]https://www.topcoder.com/community/data-science/data-science-tutorials/algorithm-games/
[41] https://en.wikipedia.org/wiki/Arbitrary-precision_arithmetic
[42] https://pt.wikipedia.org/wiki/Ordena%C3%A7%C3%A3o_topol%C3%B3gica
[43] http://www.geeksforgeeks.org/topological-sorting/
[44] https://en.wikipedia.org/wiki/Kosaraju%27s_algorithm
[45] http://codeforces.com/blog/entry/16205
[46]https://www.topcoder.com/community/data-science/data-science-tutorials/introduction-to-graphs-and-their-data-structures-section-3/
[47] https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm
[48] http://www.tutorialspoint.com/data_structures_algorithms/spanning_tree.htm
[49] http://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/kruskal.html
[50] https://en.wikipedia.org/wiki/Eulerian_path
[51]https://www.topcoder.com/community/data-science/data-science-tutorials/a-bit-of-fun-fun-with-bits/
[52] http://codeforces.com/blog/entry/18169
[53] http://codeforces.com/blog/entry/337
[54] http://fusharblog.com/solving-linear-recurrence-for-programming-contest/
[55] http://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array/
[56]https://www.topcoder.com/community/data-science/data-science-tutorials/maximum-flow-section-2/
[57] http://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/
[58] http://www.geeksforgeeks.org/bridge-in-a-graph/
[59] https://community.topcoder.com/tc?module=Static&d1=features&d2=082803

[60] http://www.facom.ufu.br/~madriana/EBD/Pilha.pdf
[61] https://pt.wikipedia.org/wiki/Lista_ligada
[62] http://www2.ic.uff.br/~boeres/slides_ed/ed8.pdf
[63] https://en.wikipedia.org/wiki/Binary_heap
[64]https://www.topcoder.com/community/data-science/data-science-tutorials/disjoint-set-data-structures/
[65]https://www.topcoder.com/community/data-science/data-science-tutorials/binary-indexed-trees/
[66]https://www.topcoder.com/community/data-science/data-science-tutorials/range-minimum-query-and-lowest-common-ancestor/
[67] http://codeforces.com/blog/entry/15729
[68] http://codeforces.com/blog/entry/15890
[69] http://e-maxx.ru/algo/segment_tree
[70] https://en.wikipedia.org/wiki/Treap
[71]https://www.topcoder.com/community/data-science/data-science-tutorials/using-tries/
[72]https://www.topcoder.com/community/data-science/data-science-tutorials/geometry-concepts-basic-concepts/
[73] https://www.quora.com/What-is-coordinate-compression