



LISTA DE CONTEÚDOS DO NÍVEL UNIVERSITÁRIO

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

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

4. Conceitos básicos de Matemática Discreta

- [J] Indução matemática; [6]
- [J] Relações (reflexão, simetria, ordem lexicográfica, etc); [7]
- [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] Árvores 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] Exponenciação 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 $O(N \log N)$: heap sort, merge sort, etc; [\[34\]](#) [\[35\]](#)
- j. [N2] Operações simples em inteiros de tamanho arbitrário; [\[41\]](#)
- k. [N2] Algoritmos de força bruta e programação dinâmica com auxílio de máscaras de bits; [\[51\]](#) [\[52\]](#) [\[53\]](#)
- l. [N2] Exponenciação de matrizes para resolver problemas de programação dinâmica; [\[54\]](#)

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; [\[50\]](#)

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 $O(\log N)$. [\[66\]](#)
- j. [N2] Árvore de Fenwick (binary indexed tree) 2D; [\[65\]](#)
- k. [N2] Árvore de segmentos (Segment tree); [\[67\]](#) [\[68\]](#) [\[69\]](#)

10. Geometria computacional

- a. [N1] Pontos, vetores, linhas e segmentos de linhas; [\[72\]](#)
- b. [N1] Pontos colineares, vetores paralelos e ortogonais. [\[72\]](#)
- c. [N1] Interseção de duas linhas; [\[72\]](#)
- d. [N2] Compressão de coordenadas; [\[73\]](#)
- e. [N2] Convex hull em $O(N \log N)$; [\[72\]](#)
- f. [N2] Line sweep; [\[4\]](#)