**NỘI DUNG CƠ BẢN CẦN THIẾT NÊN NẮM**

Vì đây chỉ là post tóm tắt, nên mình sẽ chỉ tóm tắt tên các thuật (hoặc kỹ thuật) cơ bản nên biết, không nói quá sâu và chi tiết vào bất kỳ thuật (hoặc kỹ thuật) nào.

Đây chỉ là phần liệt kê nội dung cơ bản nên mình sẽ không đề cập tới các kỹ thuật quá cao siêu như các kỹ thuật tối ưu DP siêu cấp vip pro (Knuth Optimization, Slope Trick v.v các kiểu) và các CTDL, kiến thức căng cực.

**I : Đại số**

1/ [Mũ nhanh (Binary Exponentiation)](https://cp-algorithms.com/algebra/binary-exp.html)

2/ [Sàng nguyên tố](https://cp-algorithms.com/algebra/sieve-of-eratosthenes.html) (thuật [Sàng O(N)](https://cp-algorithms.com/algebra/prime-sieve-linear.html))

3/ [Phi hàm Euler](https://cp-algorithms.com/algebra/phi-function.html)

4/ Định lý Fermat nhỏ

5/ [Nghịch đảo modulo](https://cp-algorithms.com/algebra/module-inverse.html)

**II : Tổ hợp**

1/ [Tổ hợp, hoán vị](https://cp-algorithms.com/combinatorics/binomial-coefficients.html)

2/ [Bao hàm loại trừ](https://vnoi.info/wiki/translate/he/Number-Theory-7.md)

**III/ [CẤU TRÚC DỮ LIỆU](https://vnoi.info/wiki/algo/data-structures/data-structures-overview.md)**

1/ [Segment Tree](https://vnoi.info/wiki/algo/data-structures/segment-tree-extend.md) (IT)/[Fenwick Tree](https://vnoi.info/wiki/algo/data-structures/fenwick.md) (BIT) (hoặc bài của [cp-algo](https://cp-algorithms.com/data_structures/fenwick.html) cũng khá ổn)

2/ [Deque tìm min/max trên đoạn tịnh tiến](https://vnoi.info/wiki/algo/data-structures/deque-min-max.md)

3/ [Chia căn (Sqrt Decomposition)](https://vnoi.info/wiki/algo/data-structures/sqrt-decomposition.md)

4/ [Sparse table](https://vnoi.info/wiki/translate/topcoder/Range-Minimum-Query-and-Lowest-Common-Ancestor.md)

\* Lưu ý : đối với CTDL Segment Tree, mình khuyên mọi người nên học bản Segment Tree + Đệ quy (với bộ nhớ 4\*N) hơn là bản Segment Tree + khử đệ quy (với bộ nhớ 2\*N).  
**IV : Đồ thị**

1/ Duyệt đồ thị : BFS và DFS

2/ Cầu khớp, liên thông mạnh v.v : lý thuyết, thuật toán thì mình recommend Tarjan vì mình chỉ biết mỗi Tarjan :| (sách CP3[1])

3/ Đường đi ngắn nhất (DDNN):

- Một đỉnh nguồn (Single-Source Shortest Path) : Dijkstra (trang 146 sách CP3), [BFS 0-1](https://cp-algorithms.com/graph/01_bfs.html)

- Kiểm tra chu trình có chu trình âm : [Bellman-Ford](https://cp-algorithms.com/graph/bellman_ford.html) (hoặc có thể đọc trang 151 sách CP3)

- Mọi đỉnh nguồn (All-pairs Shortest Path) : Floyd-Warshall (trang 155 sách CP3)

\* Lưu ý : Đối với đồ thị không có trọng số, thuật toán BFS có thể được dùng để thay thế Dijkstra trong việc tìm DDNN từ đỉnh nguồn.

4/ Cây khung : Prim và Kruskal (recommend nên xài Kruskal hơn vì dễ code, dễ debug)

\* Lưu ý : Kruskal luôn đi kèm với Disjoint Set

5/ [Lowest Common Ancestor (LCA)](https://vnoi.info/wiki/translate/topcoder/Range-Minimum-Query-and-Lowest-Common-Ancestor.md)

6/ Luồng (Max flow) : Edmond Karp, Dinic (không recommend), …

**V : Geometry (Đọc chương 7 CP3)**

**VI : [Xử lý chuỗi](https://vnoi.info/wiki/algo/string/basic.md)**

1/ [String hashing](https://vnoi.info/wiki/algo/string/hash.md)

2/ [cây Trie](https://leduythuccs.github.io/2018-05-23-c-u-tr-c-d-li-u-trie/)

**VII : [Quy hoạch động](https://vnoi.info/wiki/translate/topcoder/dynamic-programming.md)**

Các bài về QHĐ của VNOI

**VIII : Khác**

1/ Rời rác hóa : thường được dùng để nén giá trị số xuống đủ nhỏ để các CTDL như Fenwick Tree (BIT) có thể chạy được.

2/ Duyệt phân tập (Meet in middle)

3/ [Tìm kiếm nhị phân (Binary Search)](https://codeforces.com/edu/course/2/lesson/6), mình recommend mọi người nên đấm phần course Binary Search của CF Edu

4/ [Hai con trỏ (2-pointer)](https://codeforces.com/edu/course/2/lesson/9), mình cũng khuyến khích mọi người nên đấm phần 2 Pointer của CF EDU

5/ [Disjoint Set (DSU)](https://codeforces.com/edu/course/2/lesson/7), cũng như trên, khuyến khích nên đấm DSU của CF EDU

…

Các source học tập bổ ích : Ngoài phần tóm tắt nội dung kiến thức nên nắm trên, mình cũng recommend mọi người một vài source học tập cũng bổ ích

- [CSES Problem Set](https://cses.fi/problemset/list/)

- [A2OJ Ladders](https://a2oj.com/Ladders.html) (dành cho các thanh niên có hứng tryhard rank cao CF)

- [Codeforces ITMO pilot course](https://codeforces.com/edu/course/2)

- [CP Algo](https://cp-algorithms.com/) (trang này có nhiều bài dịch từ Emaxx)

- [VNOI](https://vnoi.info/wiki/Home) (tất nhiên không thể thiếu)

- [USACO Guide](https://usaco.guide/dashboard/) (dành cho người muốn tryhard đấm mặt các idol quốc tế)

- Competitive Programming 3

- [Competitive Programming Handbook](https://cses.fi/book/book.pdf)

- Sách chuyên tin của thầy Lê Minh Hoàng

- [Atcoder DP contest](https://atcoder.jp/contests/dp/tasks) (làm được 10 bài đầu tiên/24 bài là tay bạn đã rất to)

[1] : Sách CP3 là cuốn Competitive Programming 3 của Steven Halim