

## Làm lại các bài sau:

\* Đánh (\*) là những phần quan trọng.

<a href="http://vn.spoj.com/problems/FLOYD/">http://vn.spoj.com/problems/FLOYD/</a>	<b>Dijkstra*</b>
<a href="http://vn.spoj.com/problems/GRAPH_/">http://vn.spoj.com/problems/GRAPH_/</a>	Khớp cầu
<a href="http://vn.spoj.com/problems/QBMST/">http://vn.spoj.com/problems/QBMST/</a>	<b>Prim/ Kruskal*</b>
<a href="http://vn.spoj.com/problems/SUBSTR/">http://vn.spoj.com/problems/SUBSTR/</a>	<b>Hash*</b>
<a href="http://vn.spoj.com/problems/NKLINEUP/">http://vn.spoj.com/problems/NKLINEUP/</a>	<b>IT*</b> hoặc <b>RMQ</b>
<a href="http://vn.spoj.com/problems/KPLANK/">http://vn.spoj.com/problems/KPLANK/</a>	<b>LeftRight*</b>
<a href="https://vn.spoj.com/problems/SEC/">https://vn.spoj.com/problems/SEC/</a>	Trie
<a href="https://vn.spoj.com/problems/IOIBIN/">https://vn.spoj.com/problems/IOIBIN/</a>	<b>DSU*</b>
<a href="https://vn.spoj.com/problems/LCA/">https://vn.spoj.com/problems/LCA/</a>	LCA

Ngoài ra nên xem lại các kiến thức:

- + Hình học ( phương trình đoạn thẳng, khoảng cách, cách tính diện tích một hình, convexhull, ...).
- + Floyd (tìm đường đi ngắn nhất giữa mọi cặp đỉnh, mã bài FLOYD trên SPOJ)
- + Xử lí số lớn
- + Toán ( tính gcd, lcm, sàng số nguyên tố, modulo,... )
- + Nhân Ấn Độ
- + Tarjan - Tìm thành phần liên thông mạnh
- + Luồng, cặp ghép
- + Dequeue
- + KMP/ Hàm Z
- + Nhân ma trận
- + Tìm kiếm nhị phân\*
- + Hai con trỏ\*
- + DFS/BFS \*

- + Cấu trúc dữ liệu (set,map,vector...)\*
- + Chia đôi tập hợp
- + Sắp xếp Topo
- + Chặt tam phân

## **Ôn thêm các thuật toán:**

- Duyệt trâu: + Quay lui (sinh nhị phân, sinh tam phân, sinh hoán vị...).
- + Tham lam (không chính xác hoàn toàn nhưng ăn được một số test).
- + Duyệt bình thường (đề nói gì làm đó).

\* Duyệt quan trọng nhất, vào làm bài chưa nghĩ ra thuật toán tối ưu thì cứ làm duyệt để đảm bảo có điểm. Nếu nghĩ ra cách tối ưu hơn thì có thể dùng code duyệt để kiểm tra độ chính xác của code tối ưu.

## **Lưu ý khác:**

- Trước khi bắt tay vào suy nghĩ, các bạn nên đọc kỹ đề bài từ 2-3 lần, trong tầm 5-10 phút để đảm bảo k bỏ sót/hiểu nhầm đề.
- Chia thời gian làm bài hợp lí. Đây là một kĩ năng nhỏ nhưng quan trọng, trong khi làm bài, bạn luôn phải để ý lượng thời gian còn lại, để xem mình nên nghĩ tiếp một bài hay không, hay bắt đầu việc code trâu luôn, hoặc là bạn có thể đi wc thêm bao nhiêu lần nữa, ăn uống, ngủ nghỉ, chơi dò mìn... Một trong những lỗi về chiến lược mà kể cả những người đã thi nhiều lần mà vẫn mắc, đó là quá cay cú đi tìm lời giải cho một bài toán, kết quả là không còn thời gian chạy trâu các bài còn lại. Nên nhớ rằng, đôi khi sub cuối bài 2 khó hơn rất nhiều, nhưng lại ít điểm hơn sub 1 bài 3, do đó trước khi bắt đầu nghĩ một bài, bạn cần đặt giới hạn mức thời gian nghĩ bài đó.
- Dành từ **45p-60p cuối giờ** để sinh test và kiểm tra bài làm. Đừng cố gắng nghĩ bài mới vào thời gian này mà hãy kiểm tra chắc chắn bài làm của mình, không bị sai vặt, điểm sẽ cao hơn. Dù có đang làm được ít nhưng hãy nghĩ rằng mọi người cũng ít điểm như mình, vẫn cứ dừng nghĩ lại để sinh test kiểm tra.

- Cách sinh test:

- + Sinh các trường hợp đặc biệt mà mình nghĩ ra.
  - + Sinh các trường hợp bình thường, test nhỏ, dễ kiểm tra bằng tay.
  - + Có thời gian thì sinh những test lớn hơn và dùng code duyệt để so sánh với code tối ưu.
- Về các bài multitest : Khi có multitest, việc code bug hoặc quên khởi tạo sẽ tạo hậu quả rất nghiêm trọng (90% là bạn sẽ được 0 điểm). Do đó bạn cần ưu tiên sinh test và kiểm tra kĩ những bài này. Mẹo mình hay sử dụng là sinh ra một bộ test khoảng 2-4 test con bao gồm những trường hợp trên rồi lặp lại (copy - paste) 2 - 3 lần, như vậy sẽ hạn chế được việc quên khởi tạo lại mảng/ biến/ ...
- Lỗi vặt dễ xảy ra:
- + Khai báo mảng thiếu.
  - + Cẩn thận số lớn (int64\_t hoặc long long thay vì dùng int).
  - + Những bài có số thực sử dụng kiểu double (tuyệt đối không sử dụng kiểu float).
  - + Hạn chế sử dụng số thực đến mức tối đa, ví dụ phải so sánh căn bậc 2 của 2 số thì có thể so sánh trực tiếp 2 số thay vì lấy căn rồi mới so sánh. Nếu buộc phải so sánh số thực thì sử dụng một epsilon rất nhỏ ( ví dụ so sánh  $x == y$ , thì so sánh  $|x-y| \leq 0.0000001$ , eps tùy thuộc vào độ chính xác mà bài toán yêu cầu).
  - + Sai tên file .CPP, .INP, .OUT (sai là 0 điểm luôn).
  - + Sử dụng nhập xuất file (lỡ quên mà xuất ra màn hình cũng 0 điểm luôn).
  - + Lưu bài cẩn thận, nếu không lỡ may cúp điện sẽ mất bài.
- Lưu ý khác:
- + Một số máy nếu xui sẽ bị ẩn đuôi file, rất bất tiện trong việc tạo file .INP, nên học trước cách làm hiện đuôi file.  
(<https://blogchiasekienthuc.com/thu-thuat-may-tinh/hien-thi-duoi-mo-rong-cua-cac-file-tren-windows.html>)
  - + Khi nộp bài, em chỉ cần lấy những bài đuôi file là .CPP để nộp (không được nộp nguyên folder).
  - + Khi làm bài xong, các em nên giấu bài vào một nơi nào đó để lúc giám khảo xóa bài gốc thì còn bài của mình (giấu bằng cách ẩn file đó đi, đừng giấu vào ổ

C: vì ổ C: bị đóng băng).

+ Thường thì các em được vào phòng thi trước 30p, nên hãy tận dụng thời gian để test máy (bàn phím, chuột..., nếu khó bấm chuột hoặc bàn phím bị hỏng thì yêu cầu ngay giám thị đổi cái khác), code thử một số bài đơn giản, chạy chương trình, debug để coi thử codeblocks bị lỗi gì không. Nếu còn thời gian, em chuẩn bị một số thuật toán trước, nếu trong bài thi có sử dụng thuật toán đó thì em sẽ lợi thời gian.

+ Mang theo nước, đồ ăn, hỏi giám thị, nếu được phép thì cứ mang vào phòng thi.