LÝ THUYẾT TRÒ CHƠI

Trại hè 2018 – Hạ Long

Phạm Bá Thái - ĐH Bách Khoa Hà Nội
thai9cdb@gmail.com

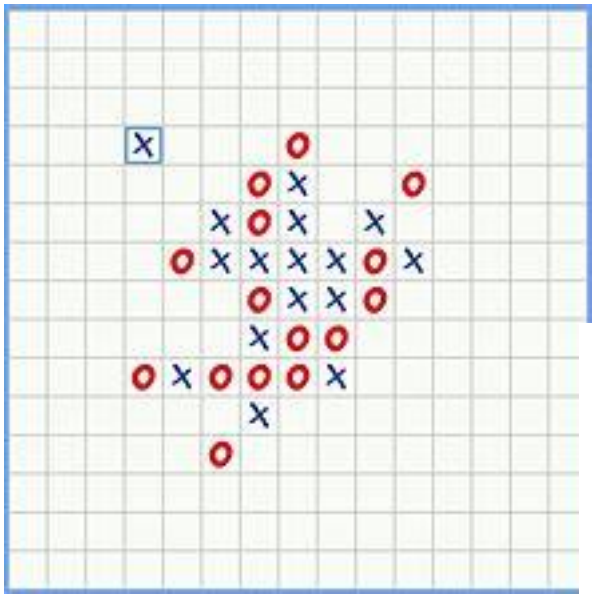
Nội dung

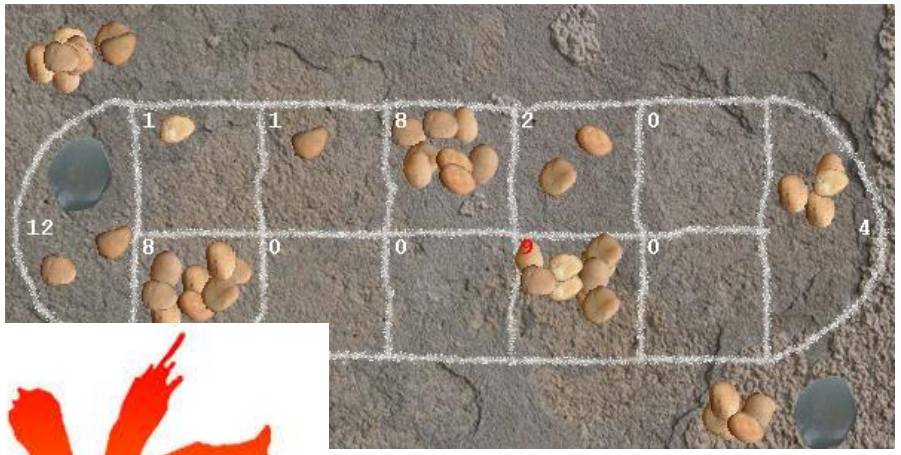
- Phần A: Cơ bản
  - I. Trò chơi đối kháng đơn giản
  - II. Chiến thuật N-P
  - III. Sắp xếp topo và ứng dụng
  - IV. Đệ quy có nhớ và ứng dụng
- Phần B: Nâng cao
  - V. Hàm Grundy trên DAG
  - VI. Tổng trò chơi

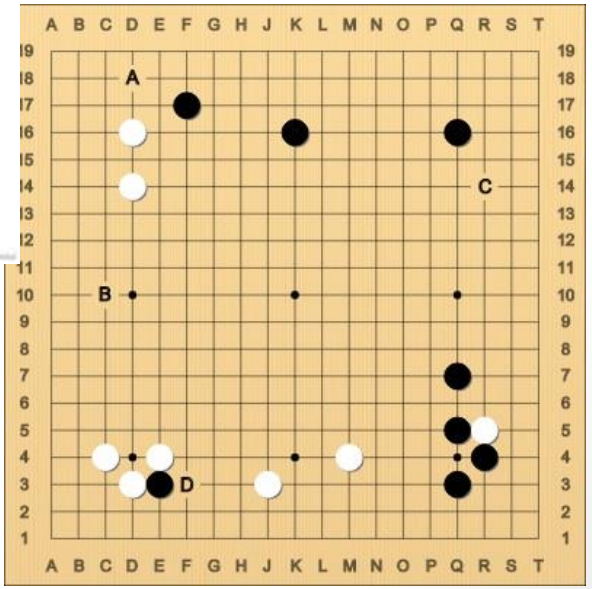
PHẦN A: CƠ BẢN

Trại hè 2018 – Hạ Long

I. Trò chơi đối kháng đơn giản





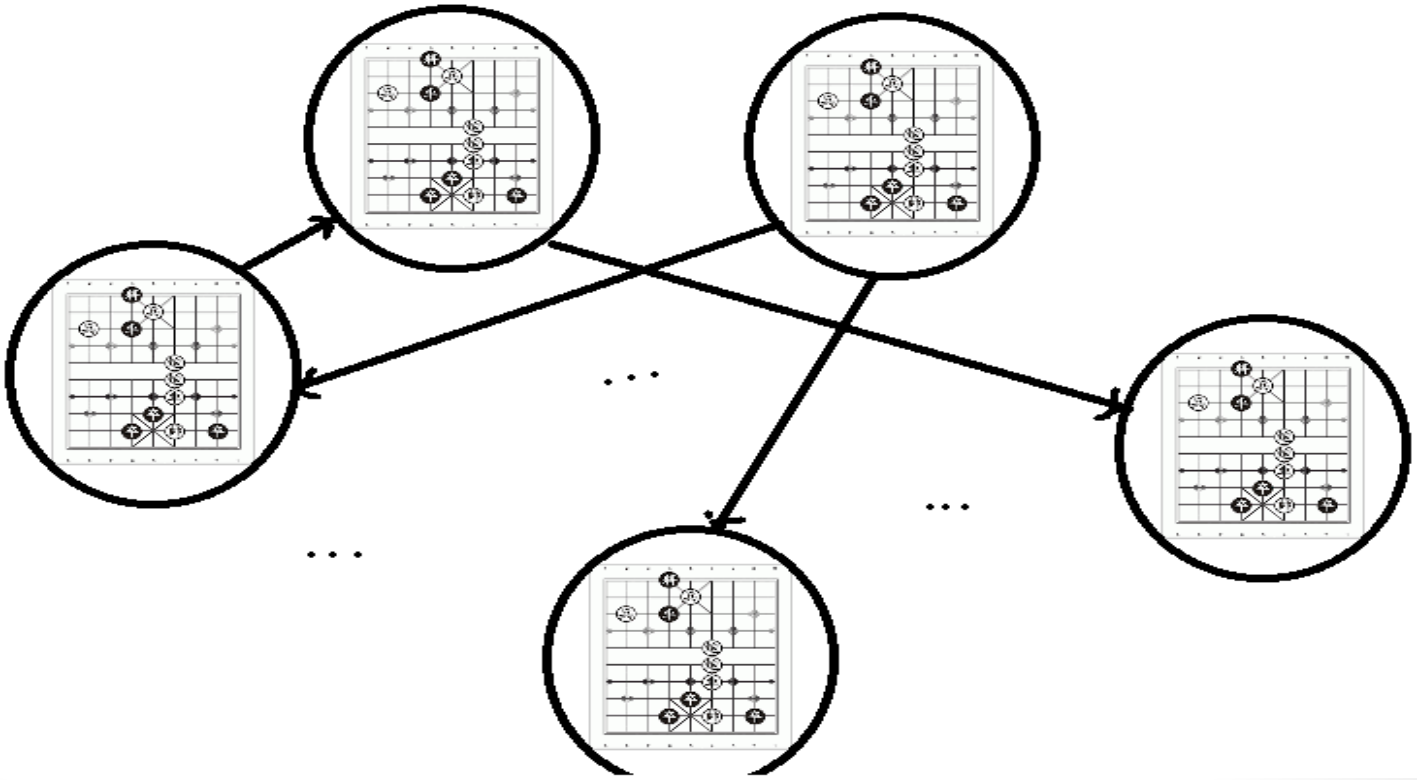


I. Trò chơi đối kháng đơn giản

- Trò chơi đối kháng:
  - Tập khác rỗng các trạng thái của trò chơi
  - Tập trạng thái con khác rỗng các trạng thái kết thúc
  - Tập các luật chơi (nước đi) hợp lệ
  - Có hai người chơi, luân phiên nhau thực hiện một nước đi hợp lệ

I. Trò chơi đối kháng đơn giản

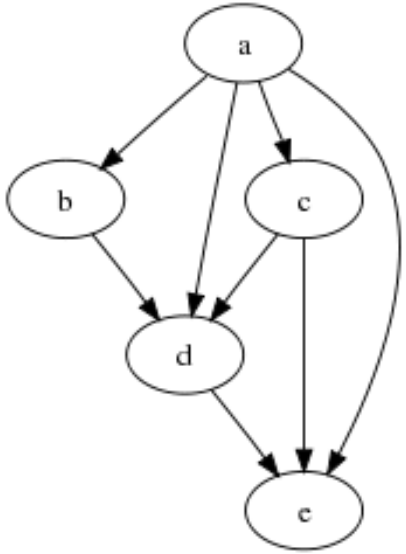
- Mô tả trò chơi đối kháng bằng đồ thị có hướng:
  - Mỗi trạng thái của trò chơi là một đỉnh của đồ thị
  - Mỗi luật chơi (nước đi) hợp lệ là một cung (nối trạng thái trước và sau khi thực hiện nước đi)



I. Trò chơi đối kháng đơn giản

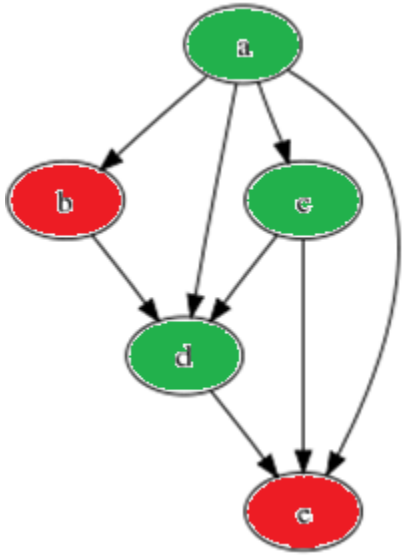
- Trò chơi đối kháng đơn giản (hay trò chơi tổ hợp cân bằng):
  - Là trò chơi đối kháng
  - Không lặp lại các trạng thái đã đi qua
  - Tập kết thúc là tập các đỉnh không có cung đi ra

  => Đồ thị có hướng không có chu trình (DAG)



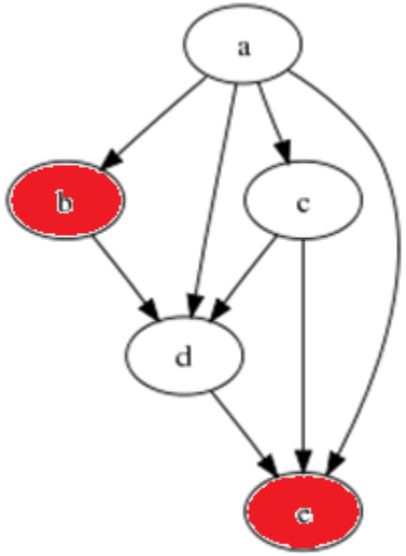
II. Chiến thuật N-P

- Tập các trạng thái được phân hoạch thành hai tập
  - P: Tập các trạng thái thua (màu đỏ)
  - N: Tập các trạng thái thắng (màu xanh)
  - Với trò chơi đối kháng đơn giản, không có trạng thái hòa, cũng không có trạng thái nào là vừa thắng vừa thua. Lúc này N và P tạo thành một phân hoạch của tập các trạng thái



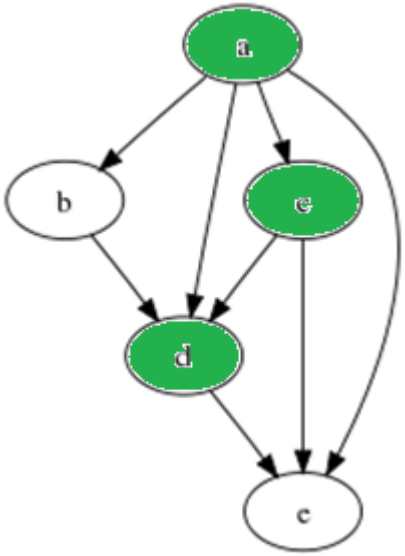
II. Chiến thuật N-P

- Các tính chất của tập P:
  - Tập các trạng thái kết thúc là tập con của P
  - Từ một trạng thái thuộc P, hoặc không thể di chuyển nữa, hoặc chỉ di chuyển được sang các trạng thái thuộc N
  - Nếu người chơi đang ở trạng thái thuộc tập P, dù họ đi thế nào thì đối thủ của họ vẫn có chiến thuật để chiến thắng



II. Chiến thuật N-P

- Các tính chất của tập N:
  - N là phần bù của P trong tập các trạng thái
  - Từ một trạng thái thuộc N, tồn tại một nước đi để chuyển sang trạng thái thuộc P
  - Nếu người chơi đang ở trạng thái thuộc tập N, họ có chiến thuật chiến thắng

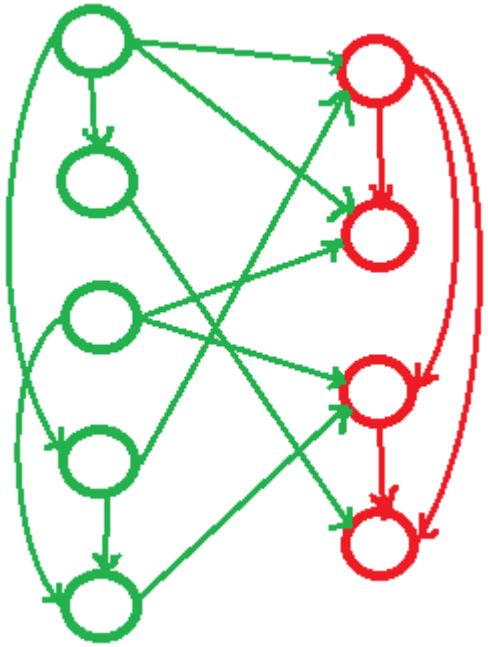


II. Chiến thuật N-P

- Thuật toán tìm N và P:
  - N := Ø, P := {Tập các trạng thái kết thúc}
  - Lặp lại
    - N := N ∪ {Tập các trạng thái có thể đến được P}
    - P := P ∪ {Tập các trạng thái mà mọi nước đi đều đi sang N}
  - Cho đến khi N ∪ P = {Tập trạng thái ban đầu}
- Thuật toán trên dừng, vì sao?

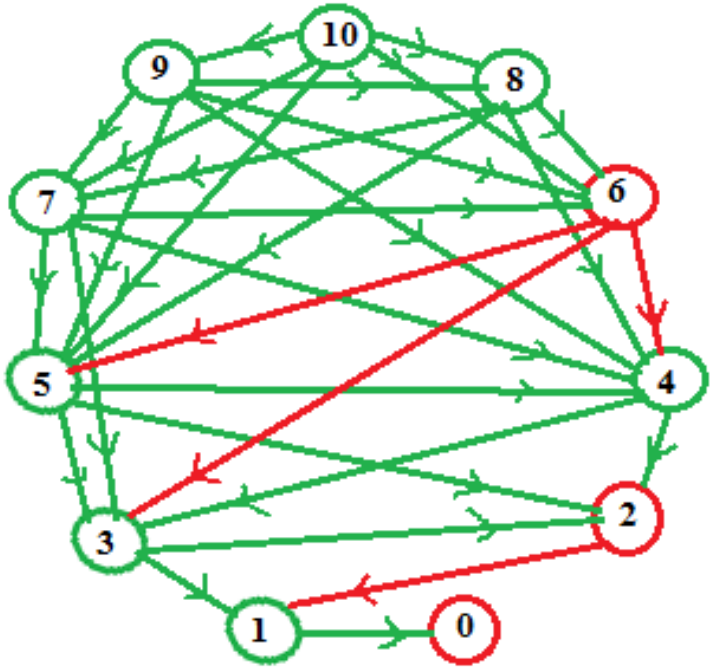
II. Chiến thuật N-P

- Chiến thuật chơi thắng cho người chơi ở trạng thái thuộc N
  - Đi sang một trạng thái thuộc P, nước đi này là tồn tại
  - Đối thủ hoặc bị thua, hoặc thực hiện được một di chuyển hợp lệ nhưng quay lại tập N
  - Lặp lại chiến thuật trên
- Chiến thuật trên dừng và chiến thắng, vì sao?



II. Chiến thuật N-P

- Trò chơi ví dụ: Hai người chơi một trò chơi với n viên bi như sau:
  - Hai người luân phiên nhau thực hiện lượt chơi
  - Mỗi lượt chơi, cần lấy đi một ít bi, ít nhất là một viên và nhiều nhất là (k+1)/2 với k là số bi hiện tại
  - Ai lấy được viên bi cuối cùng là người thắng cuộc
- Hình sau mô tả đồ thị trò chơi với n=10

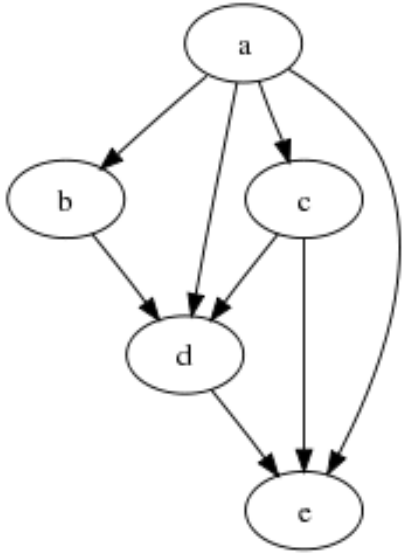


III. Sắp xếp topo và ứng dụng

- Cho một tập các công việc cần phải làm, trong đó có một số việc buộc phải làm sau một số việc khác. (Ví dụ bạn không thể mang một cái xe đi bán nếu cái xe đó còn chưa được sản xuất)
- Các công việc không mâu thuẫn nhau, khi đó tồn tại một kế hoạch làm việc (thứ tự thực hiện các công việc) thỏa mãn các ràng buộc

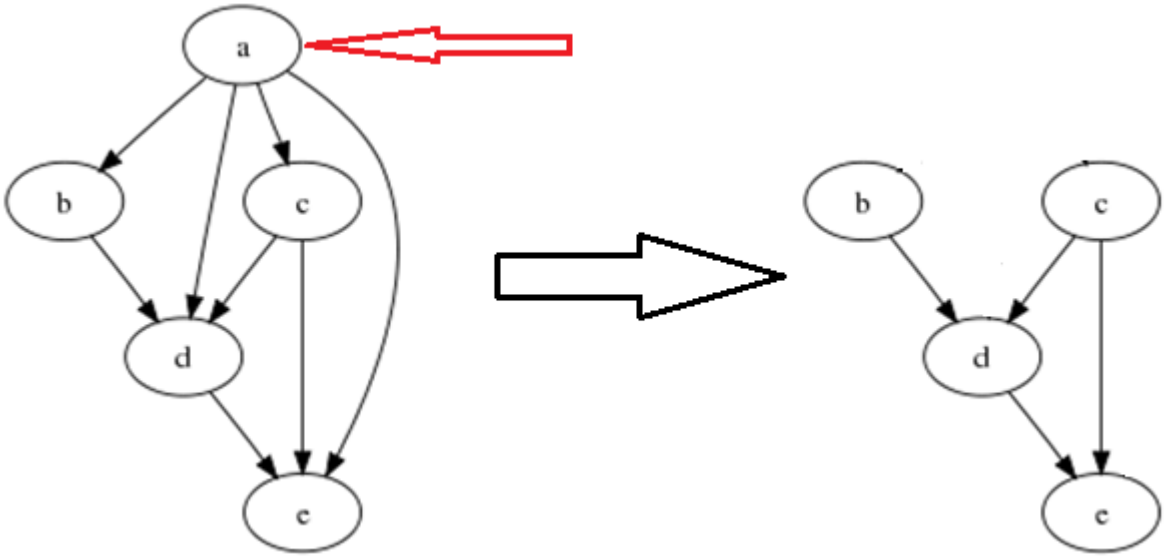
III. Sắp xếp topo và ứng dụng

- Đồ thị mô tả các công việc:
  - Tập đỉnh là tập các công việc
  - Mỗi cung nối x với y cho biết x phải làm trước y
  - Các công việc không mâu thuẫn nhau được thể hiện ở việc đồ thị tạo thành không có chu trình (DAG)



III. Sắp xếp topo và ứng dụng

- Một đồ thị có hướng không có chu trình (DAG) thì:
  - Tồn tại một đỉnh không có cung đi vào (?)
  - Sau khi xóa một đỉnh, đồ thị thu được vẫn là DAG (?)

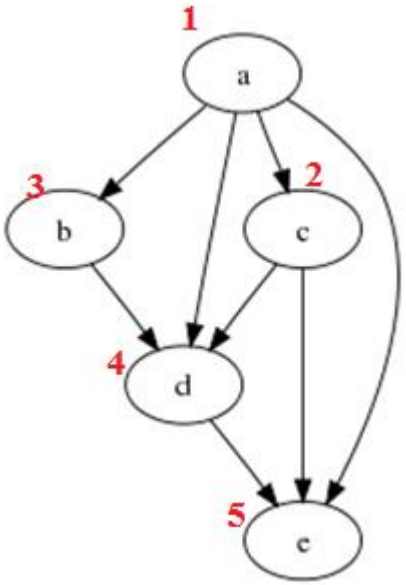


III. Sắp xếp topo và ứng dụng

- Thuật toán tìm thứ tự topo trên DAG:
  - Queue T := Ø
  - Lặp lại
    - Tìm x không có cung đi vào (tồn tại x như vậy)
    - Xóa x (và các cung kề x) ra khỏi đồ thị
    - T.push(x)
  - Cho đến khi mọi đỉnh đều ở trong T
  - Đưa ra T là một thứ tự topo

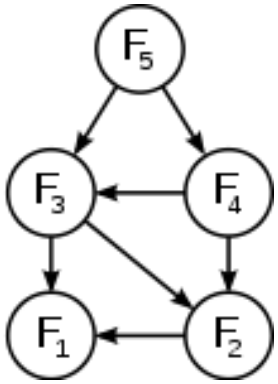
III. Sắp xếp topo và ứng dụng

- Trên DAG:
  - Luôn tồn tại thứ tự topo
  - Thứ tự topo đó có thể không duy nhất
  - Thứ tự topo thỏa mãn các ràng buộc về thứ tự thực hiện các công việc



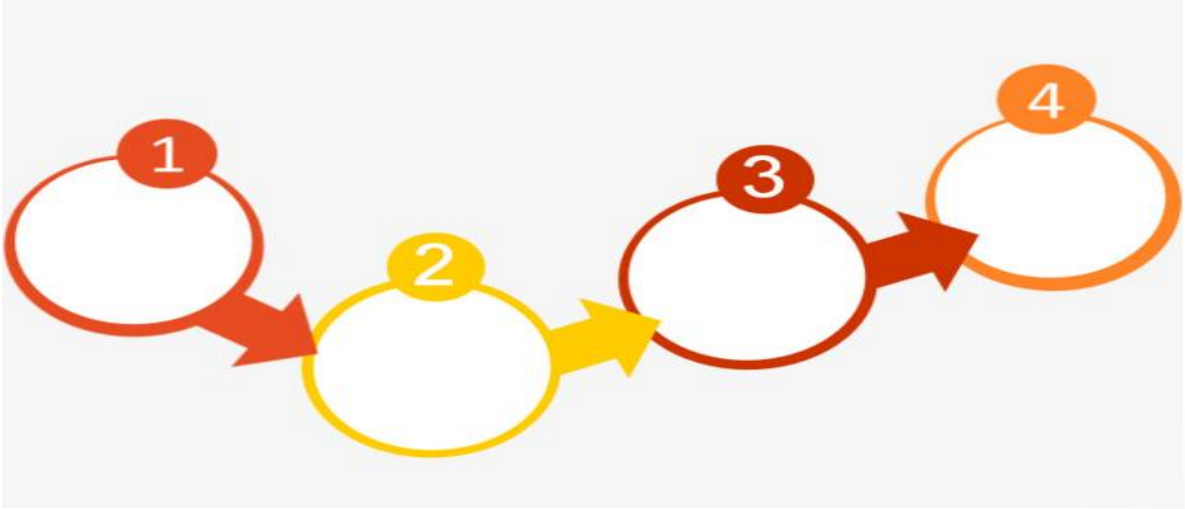
III. Sắp xếp topo và ứng dụng

- Nhiều bài toán quy hoạch động có thứ tự tính toán không tầm thường, chỉ cần các bài toán không phụ thuộc vòng tròn (chu trình) thì thứ tự topo chính là thứ tự tính toán hợp lý:
  - Lập tập đỉnh: tập các bài toán con từ bài toán ban đầu (dựa vào mảng QHĐ), mỗi bài toán con là một đỉnh của đồ thị
  - Nạp các cung (có được từ công thức QHĐ) vào đồ thị
  - Tìm một thứ tự topo trên đồ thị
  - Giải các bài toán theo thứ tự topo vừa tìm được, thứ tự giải lúc này sẽ đảm bảo công thức QHĐ chỉ gọi đến các bài toán đã được giải



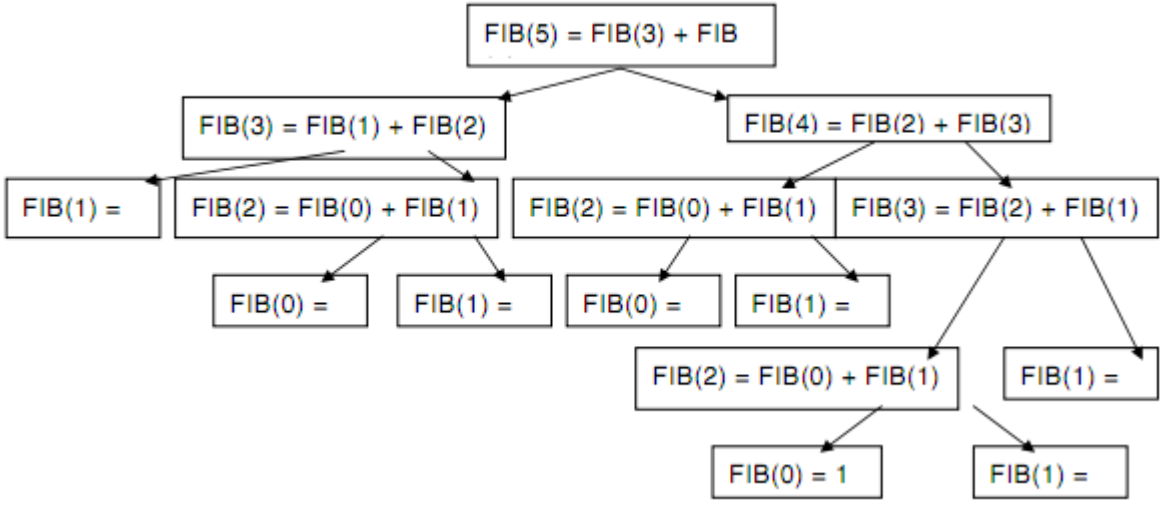
III. Sắp xếp topo và ứng dụng

- Việc phân hoạch tập các trạng thái của một trò chơi vào P và N cũng là một bài toán QHĐ, do đó ta phân các trạng thái vào hai tập kia theo thứ tự topo là hợp lý
- Ở đây đồ thị của quá trình QHĐ có các cung ngược với đồ thị của trò chơi, nên ta có thể gọi thứ tự tính toán trong trường hợp này là thứ tự topo ngược
- Việc nêu ra một thứ tự tính toán hợp lý (thứ tự topo) cũng ngầm chứng minh được rằng, mọi trò chơi đối kháng đơn giản đều là trò chơi có chiến thuật tuyệt đối



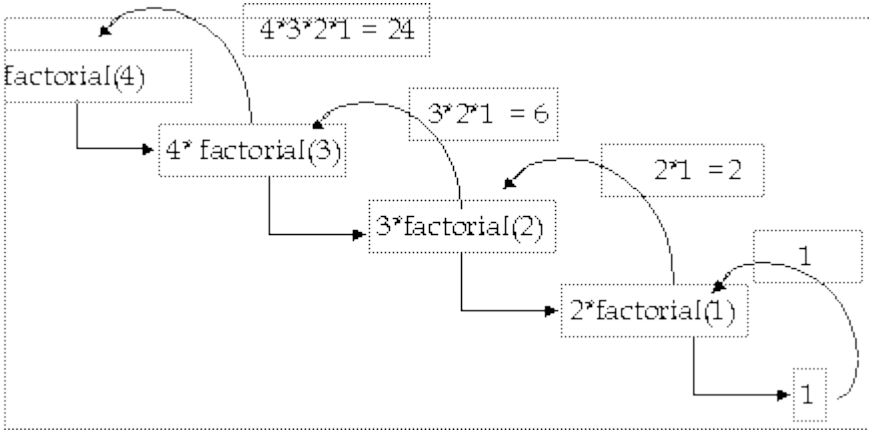
IV. Đệ quy có nhớ và ứng dụng

- Ta đã biết, mọi bài toán QHĐ đều có thể được cài đặt bằng phương pháp đệ quy có nhớ (dp). Phương pháp cài đặt này tỏ ra khá hiệu quả đặc biệt là với các bài QHĐ mà công thức của nó không chỉ ra một thứ tự tính toán tầm thường
- Dp không yêu cầu biết trước thứ tự giải các bài toán, chỉ cần các bài toán không phụ thuộc vòng tròn là được



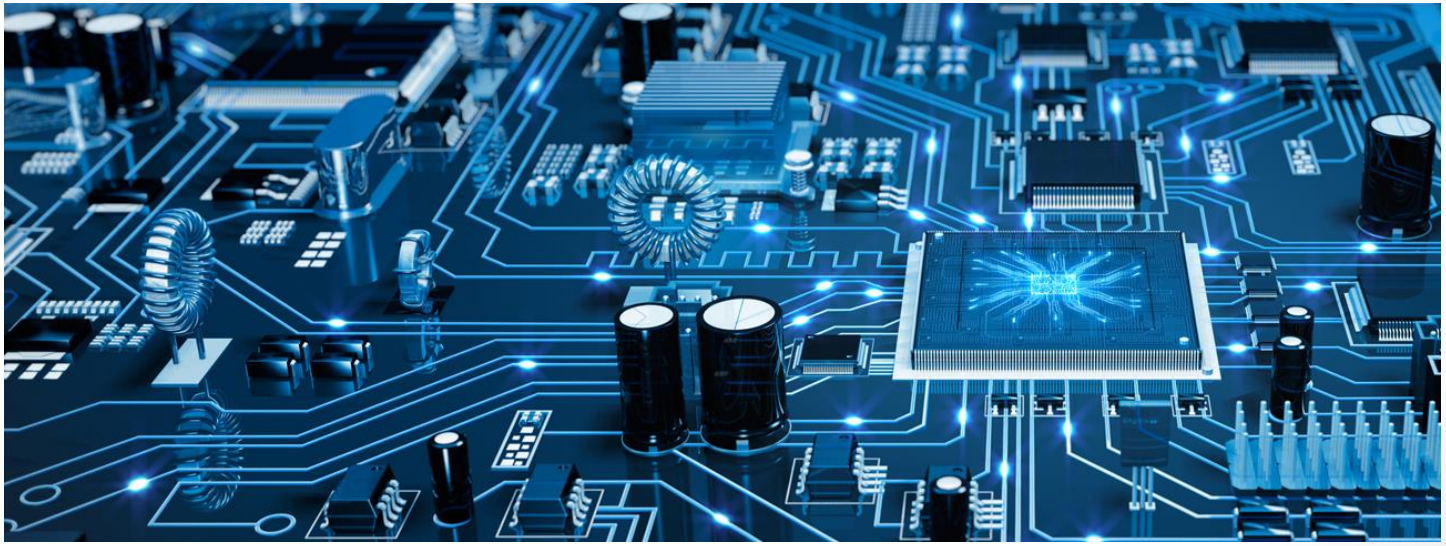
IV. Đệ quy có nhớ và ứng dụng

- Phương pháp dp tiếp cận giải bằng cách gọi đệ quy để tính:
  - Nếu bài toán này đã từng giải thì trả ra lời giải của nó đã được lưu trữ trong mảng nhớ
  - Nếu không đi giải bài toán đó và lưu kết quả vào mảng nhớ
- Tham khảo các giáo trình khác để hiểu rõ hơn về dp



IV. Đệ quy có nhớ và ứng dụng

- Có thể thấy rằng, thứ tự lưu vào mảng nhớ (cũng là thứ tự mà các bài toán được giải xong) chính là một thứ tự topo ngược
- Ta không hề sắp xếp topo trước đó, nhưng để gọi đệ quy được, trình dịch đã thay ta lập kế hoạch để tính toán, và chính máy tính đã tự mình đưa ra thứ tự topo



IV. Đệ quy có nhớ và ứng dụng

- Sắp xếp topo và dp về bản chất giống nhau, tiếp cận cài đặt khác nhau. Có thể hiểu như là tiếp cận buttom-up và top-down, hoặc tính toán theo chiều rộng và tính toán theo chiều sâu
- Điểm mạnh của dp là cài đặt đơn giản, nhanh chóng. Nhưng có thể chạy chậm và tràn stack nếu gọi đệ quy quá sâu
- Điểm mạnh của sắp xếp topo là hằng số cài đặt (ảnh hưởng trực tiếp đến thời gian chạy) nhỏ và sử dụng được bộ nhớ heap
- Đối với những bài cần thứ tự tính toán không tầm thường, trừ những TH time limit quá chặt hoặc lo sợ tràn stack, chúng ta vẫn nên dùng dp

IV. Đệ quy có nhớ và ứng dụng

- Kết luận:
  - Trò chơi đối kháng đơn giản có chiến thuật tuyệt đối
  - Thứ tự topo ngược là thứ tự tính toán chiến thuật N-P hợp lý
  - Nếu thứ tự đó đơn giản ta nên for để tính, nếu không nên dùng dp thay vì phải sắp xếp topo

THANK YOU!

Trại hè 2018 – Hạ Long

Phạm Bá Thái - ĐH Bách Khoa Hà Nội
thai9cdb@gmail.com