



BÁO CÁO ĐỒ ÁN CUỐI KÌ

MÔ PHỎNG THUẬT TOÁN THÁP HÀ NỘI SỬ DỤNG NGĂN XẾP (STACK)

|  |  |
| --- | --- |
| Lớp học phần: | 181DASA240179 |
| Giảng viên: | ThS. Trần Công Tú |
| Nhóm thực hiện: | 11 |

|  |  |
| --- | --- |
| Họ và tên | Mã số sinh viên |
| Huỳnh Quốc Hoàng Vương | 17110256 |
| Bùi Thị Hồng Nhung | 17110201 |

Tp. Hồ Chí Minh - 2018

**MỤC LỤC**

[LỜI CẢM ƠN 3](#_Toc529130540)

[DANH MỤC CÁC HÌNH 4](#_Toc529130541)

[DANH MỤC CÁC BẢNG 5](#_Toc529130542)

[1. Tổng quan đồ án. 6](#_Toc529130543)

[*1.1.* *Giới thiệu tổng quan về bài toán Tháp Hà Nội.* 6](#_Toc529130544)

[1.1.1. Lịch sử hình thành. 6](#_Toc529130545)

[1.1.2. Cách giải trong thực tế. 6](#_Toc529130546)

[1.1.3. Cách giải khác với cách nhìn của khoa học máy tính. 6](#_Toc529130547)

[1.1.4. Ứng dụng . 7](#_Toc529130548)

[*1.2.* *Về đồ án Mô phỏng thuật toán Tháp Hà Nội.* 7](#_Toc529130549)

[1.2.1. Khái niệm về mô phỏng thuật toán. 7](#_Toc529130550)

[1.2.2. Tác dụng mô phỏng thuật toán. 7](#_Toc529130551)

[1.2.3. Kiến trúc một hệ thống mô phỏng thuật toán. 7](#_Toc529130552)

[1.2.4. Mô phỏng thuật toán tháp Hà Nội. 8](#_Toc529130553)

[*1.2.5.* Lựa chọn ngôn ngữ cài đặt mô phỏng. 8](#_Toc529130554)

[*1.3.* *Lí do, mục đích chọn đồ án Demo Tháp Hà Nội.* 8](#_Toc529130555)

[2. Nội dung. 8](#_Toc529130556)

[*2.1.* *Quá trình và công việc thực hiện đồ án.* 8](#_Toc529130557)

[*2.2.* *Khái quát thuật toán tháp Hà Nội bằng ngăn xếp.* 8](#_Toc529130558)

[2.2.1. Ngăn xếp (stack) là gì. 8](#_Toc529130559)

[2.2.2. Cấu trúc và đặc điểm của ngăn xếp. 8](#_Toc529130560)

[2.2.3. Ứng dụng của ngăn xếp. 8](#_Toc529130561)

[2.2.4. Thuật toán Tháp Hà Nội bằng ngăn xếp. 8](#_Toc529130562)

[*2.3.* *Thiết kế giao diện, backdrop.* 8](#_Toc529130563)

[*2.4.* *Cài đặt thuật toán và viết chương trình.* 8](#_Toc529130564)

[*2.5.* *Kiểm lỗi và debug.* 8](#_Toc529130565)

[*2.6.* *Chương trình sản phẩm của đồ án.* 8](#_Toc529130566)

[*2.7.* *Cách thức hoạt động của chương trình.* 8](#_Toc529130567)

[3. Kết luận và rút kinh nghiệm đồ án. 9](#_Toc529130568)

[*3.1.* *Bản kế hoạch hoàn thành.* 9](#_Toc529130569)

[*3.2.* *Ưu điểm.* 9](#_Toc529130570)

[*3.3.* *Nhược điểm.* 9](#_Toc529130571)

[*3.4.* *Phương án cải thiện.* 9](#_Toc529130572)

[4. Phụ lục. 9](#_Toc529130573)

[5. Tài liệu tham khảo. 9](#_Toc529130574)

# LỜI CẢM ƠN

Để hoàn thành đồ án này, trong quá trình khảo sát và thu thập, tổng hợp thông tin em đã nhận được sự giúp đỡ tận tình từ thầy cô trong khoa và các bạn. Nhân đây, cho phép em bày tỏ lòng biết ơn chân thành với các thầy cô. Đặc biệt, đối với thầy Trần Công Tú. Bởi thầy đã hướng dẫn giúp đỡ tận tình trong suốt quá trình thực hiện đề tài. Trong quá trình xây dựng chương trình và nghiên cứu tìm hiểu em gặp khá nhiều khó khăn, mặt khác do trình độ còn hạn chế và những nguyên nhân khác nên dù cố gắng song đề tài của mình không tránh khỏi những hạn chế thiếu sót.

Vì thế, em rất mong nhận được sự góp ý của các thầy cô trong trường cũng như các bạn đọc. Những ý kiến đóng góp của mọi người sẽ giúp em nhận ra hạn chế và qua đó có thêm những nguồn tư liệu mới trên con đường học tập cũng như xây dựng chương trình sau này.

Em xin chân thành cảm ơn!

# DANH MỤC CÁC HÌNH

# DANH MỤC CÁC BẢNG

NỘI DUNG

1. Tổng quan đồ án.
   1. *Giới thiệu tổng quan về bài toán Tháp Hà Nội.*
      1. Lịch sử hình thành.



*Hình 1: Tháp Hà Nội.*

Trò chơi Tháp Hà Nội có thể đã xuất hiện ở Đông Á từ thế kỷ 19 hoặc trước đó. Các đĩa được làm bằng sứ ở Trung Quốc, Nhật Bản và Việt Nam. Trò chơi này được đưa sang phương Tây lần đầu bởi nhà toán học người Pháp Edouard Lucas vào năm 1883. Trò chơi này nhanh chóng được các nhà toán học nghiên cứu sau đó, và trở nên nổi tiếng vì độ phức tạp của mình. Lời giải tối ưu cho trò chơi có thể tìm thấy chính xác cho trường hợp 3 cọc. Nhưng khi mở rộng cho 4 cọc hoặc nhiều hơn, lời giải chính xác cho đến nay vẫn chưa được khẳng định.

* + 1. Cách giải trong thực tế.

Đế đặt nằm ngang, các cột thẳng đứng. Các đĩa đặt theo thứ tự từ lớn đến nhỏ từ thấp lên cao, tạo nên một Tòa tháp. Trò chơi đòi hỏi di chuyển các đĩa, bằng cách đặt chúng vào cột bên cạnh, một đĩa trong một di chuyển, theo luật sau:

I. Sau mỗi di chuyển, các đĩa đều nằm trên một, hai, hoặc ba cột, theo thứ tự từ lớn đến nhỏ từ thấp đến cao.

II. Đĩa trên cùng của một trong ba cột đĩa được đặt vào cột rỗng.

III. Đĩa trên cùng của một trong ba cột đĩa được đặt lên một cột đĩa khác, nếu đĩa này nhỏ hơn các đĩa của cột này.

Trò chơi có thể dễ dàng tự khám phá, bằng việc giải quyết dần từ 3, 4, và 5 đĩa. Trò chơi luôn giải được và đòi hỏi thời gian chơi lâu khoảng gấp đôi mỗi khi cho thêm một đĩa vào Tòa tháp.

* + 1. Cách giải khác với cách nhìn của khoa học máy tính.

Bài toán Tháp Hà Nội trở thành ví dụ về phương pháp giải đệ quy kinh điển trong dạy học và tin học. Sau đây là dạng rút gọn của giải thuật đệ quy áp dụng giải bài toán này:

1. Chuyển đĩa 1 sang cọc C
2. Chuyển đĩa 2 sang cọc B
3. Chuyển đĩa 1 từ C sang B sao cho nó nằm lên 2.

Vậy ta hiện có 2 đĩa đã nằm trên cọc B, cọc C hiện thời trống

1. Chuyển đĩa 3 sang cọc C
2. Lặp lại 3 bước trên để chuyển 1 & 2 cho nằm lên 3

Mỗi lần dựng xong tháp từ đĩa i đến 1, chuyển đĩa i+1 từ cọc A là cọc xuất phát, rồi lại di chuyển tháp đã dựng lên đĩa i+1.

Nhưng trong đồ án này, chúng ta sẽ không sử dụng giải thuật đệ quy để giải quyết nó mà thay vào đó là dùng cấu trúc dữ liệu ngăn xếp (stack) để khử đệ quy, giải quyết bài toán.

* + 1. Ứng dụng .

Tháp Hà Nội là một bài toán thường được dùng để dạy về lập trình cơ bản. Một phiên bản bằng hình của bài toán này được lập trình trong chương trình soạn thảo *EMACS*, có thể truy cập được bằng cách gõ M-x hanoi. Ngoài ra cũng có một thuật giải mẫu viết bằng ngôn ngữ *Prolog*.

Bài toán Tháp Hà Nội thường được dùng trong nghiên cứu tâm lý về cách giải quyết vấn đề. Cũng có những biến thể khác của bài toán này gọi là Tháp Luân Đôn dùng trong chẩn đoán và điều trị thần kinh tâm lý đối với các chức năng thực hành.

* 1. *Về đồ án Mô phỏng thuật toán Tháp Hà Nội.*
     1. Khái niệm về mô phỏng thuật toán.

Mô phỏng thuật toán là quá trình tách dữ liệu, thao tác, ngữ nghĩa và mô phỏng đồ họa cho quá trình trên.

* + 1. Tác dụng mô phỏng thuật toán.

Một chương trình máy tính chứa các cấu trúc dữ liệu của thuật toán mà nó thực thi. Trong quá trình thực thi chương trình, các giá trị dữ liệu này bị thay đổi. Mô phỏng thuật toán sử dụng biểu diễn đồ họa, âm thanh để biểu diễn dữ liệu và chỉ ra sự thay đổi giá trị của dữ liệu đó trong mỗi trạng thái. Từ đó, người dùng có thể xem từng bước chương trình được thực thi và nhờ thế có thể hiểu chi tiết và đánh giá thuật toán.

* + 1. Kiến trúc một hệ thống mô phỏng thuật toán.

Đa số các hệ thống, chương trình mô phỏng thuật toán đều có những thư viện hỗ trợ thủ tục mô phỏng và giao diện mô phỏng. Nhìn chung, mỗi hệ thống, chương trình mô phỏng đều gồm 2 phần chính:

1. Phần xử lí: Xứ lí dữ liệu đầu vào của thuật toán, chạy thuật toán cần mô phỏng đã được sửa lại với mục đích gửi dữ liệu đã tính toán tới phần thứ 2.
2. Phần khung nhìn (cảnh quan): Là nơi người dùng nhìn những đối tượng mô phỏng. Thông điệp bao gồm thông tin đồ họa của đối tượng cần mô phỏng. Sau khi khung nhìn nhận dữ liệu, nó tính toán lại và kéo (render) đối tượng mức đồ họa.



Hình 2: Sơ đồ khối hệ thống mô phỏng thuật toán.

* + 1. Mô phỏng thuật toán tháp Hà Nội.

Cũng tuân theo kiến trúc trên, chương trình sản phẩm của đồ án mô tháp Hà Nội sử dụng stack của chúng em cũng tuân theo hai phần chính. Chương trình nhận dữ liệu đầu vào là số đĩa và tốc độ mô phỏng, qua quá trình tính toán, chương trình vẽ ra trên giao diện người dùng từng bước đi cụ thể của từng chiếc đĩa, đồng thời, ghi lại nhật kí (log) cho từng bước di chuyển dưới dạng kí tự (text display).

* + 1. Lựa chọn ngôn ngữ cài đặt mô phỏng.

Vì chương trình mô phỏng của chúng em biểu diễn thuật toán dưới dạng các đối tượng có thuộc tính, hành vi rõ ràng nên nhóm chúng em quyết định sử dụng ngôn ngữ Microsoft C#.NET với phiên bản .NET Framework 4.5.2; IDE và compiler nằm trong bộ Microsoft Visual Studio 2017 để cài đặt mô phỏng này.

* 1. *Lí do, mục đích chọn đồ án Demo Tháp Hà Nội.*

Cấu trúc dữ liệu là một chương trình bao gồm các thuật toán như sắp xếp, lựa chọn, đệ quy, ngăn xếp.Mỗi thuật toán đều có một độ khó riêng, đòi hỏi khả năng hiểu rõ thuật toán thật chính xác và có sự liên tưởng thật phong phú để làm sao giúp người học hiểu thật rõ về thuật toán đó.Trong phần này tôi sẽ nghiên cứu về Ngăn xếp vì để học và muốn tìm hiểu thật chắc về Ngăn xếp thì bạn phải hiểu được cách nó chạy và cách nó thực thi như thế nào. Nhờ việc mô phỏng mà việc học một ngôn ngữ hay một thuật toán sẽ dễ dàng hơn.Giúp cho quá trình dạy và học trở nên đơn giản hơn rất nhiều.Chính vì vậy chúng tôi quyết định đi xây dựng thuật toán, cụ thể là mô phỏng đề tài Demo Tháp Hà Nội.

1. Nội dung.
   1. *Khái quát thuật toán tháp Hà Nội bằng ngăn xếp.*
      1. Ngăn xếp (stack) là gì.

Trong khoa học máy tính, một ngăn xếp (còn gọi là bộ xếp chồng, tiếng Anh: stack) là một cấu trúc dữ liệu trừu tượng hoạt động theo nguyên lý "vào sau ra trước" (Last In First Out (LIFO).

* + 1. Cấu trúc và đặc điểm của ngăn xếp.

Một ngăn xếp là một cấu trúc dữ liệu dạng thùng chứa (container) của các phần tử (thường gọi là các nút (node)) và có hai phép toán cơ bản: push and pop. Push bổ sung một phần tử vào đỉnh (top) của ngăn xếp, nghĩa là sau các phần tử đã có trong ngăn xếp. Pop giải phóng và trả về phần tử đang đứng ở đỉnh của ngăn xếp. Trong stack, các đối tượng có thể được thêm vào stack bất kỳ lúc nào nhưng chỉ có đối tượng thêm vào sau cùng mới được phép lấy ra khỏi stack. Ngoài ra, stack cũng hỗ trợ một số thao tác khác:

* isEmpty(): Kiểm tra xem stack có rỗng không.
* Top() (Hay peek()): Trả về giá trị của phần tử nằm ở đầu stack mà không hủy nó khỏi stack. Nếu stack rỗng thì lỗi sẽ xảy ra.
  + 1. Ứng dụng của ngăn xếp.

Ngăn xếp có nhiều ứng dụng trong khoa học máy tính. Mà quan trọng nhất là sử dụng để khử đệ quy và quản lí bộ nhớ khi thi hành chương trình bằng cách biểu diễn các thanh ghi (register) dưới dạng ngăn xếp.

* + 1. Thuật toán Tháp Hà Nội bằng ngăn xếp.
       1. Source thuật toán Tháp Hà Nội bằng ngăn xếp bằng C#:

public struct ThuTuc

{

public int N;

public Stack<PictureBox> A;

public Stack<PictureBox> B;

public Stack<PictureBox> C;

};

public void HNTByStack(int x)

{

ThuTuc X = new ThuTuc();

X.N = x;

X.A = disksRodA;

X.C = disksRodB;

X.B = disksRodC;

Stack<ThuTuc> myStack = new Stack<ThuTuc>();

ThuTuc temp = new ThuTuc();

ThuTuc temp1 = new ThuTuc();

myStack.Push(X);

do

{

temp = myStack.Pop();

if (temp.N == 1)

Movement(temp.A, temp.B);

else

{

temp1.N = temp.N - 1;

temp1.A = temp.C;

temp1.B = temp.B;

temp1.C = temp.A;

myStack.Push(temp1);

temp1.N = 1;

temp1.A = temp.A;

temp1.B = temp.B;

temp1.C = temp.C;

myStack.Push(temp1);

temp1.N = temp.N - 1;

temp1.A = temp.A;

temp1.B = temp.C;

temp1.C = temp.B;

myStack.Push(temp1);

} while (myStack.Count != 0);

}

* + - 1. Input và output của thuật toán.

Input: Số đĩa N;

Output: Xuất ra tuần tự các bước di chuyển N đĩa từ cột A sang cột C, lấy cột B làm cột đệm (buffer).

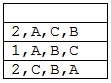
* + - 1. Minh họa cho lời gọi HNTByStack(3).

Minh họa cho lời gọi HNTByStack(3) tức là x.N = 3.

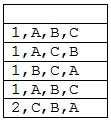
Ngăn xếp khởi đầu



Ngăn xếp sau lần lặp thứ nhất



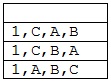
Ngăn xếp sau lần lặp thứ hai



Các lần lặp 3,4,5,6 thì chương trình con xử lý trường hợp chuyển 1 đĩa, vì vậy không có mẩu tin nào được thêm vào ngăn xếp. Mỗi lần xử lý, phần tử đầu ngăn xếp bị xoá. Ta sẽ có ngăn xếp như sau



Tiếp tục lần lặp bước 7, ta được



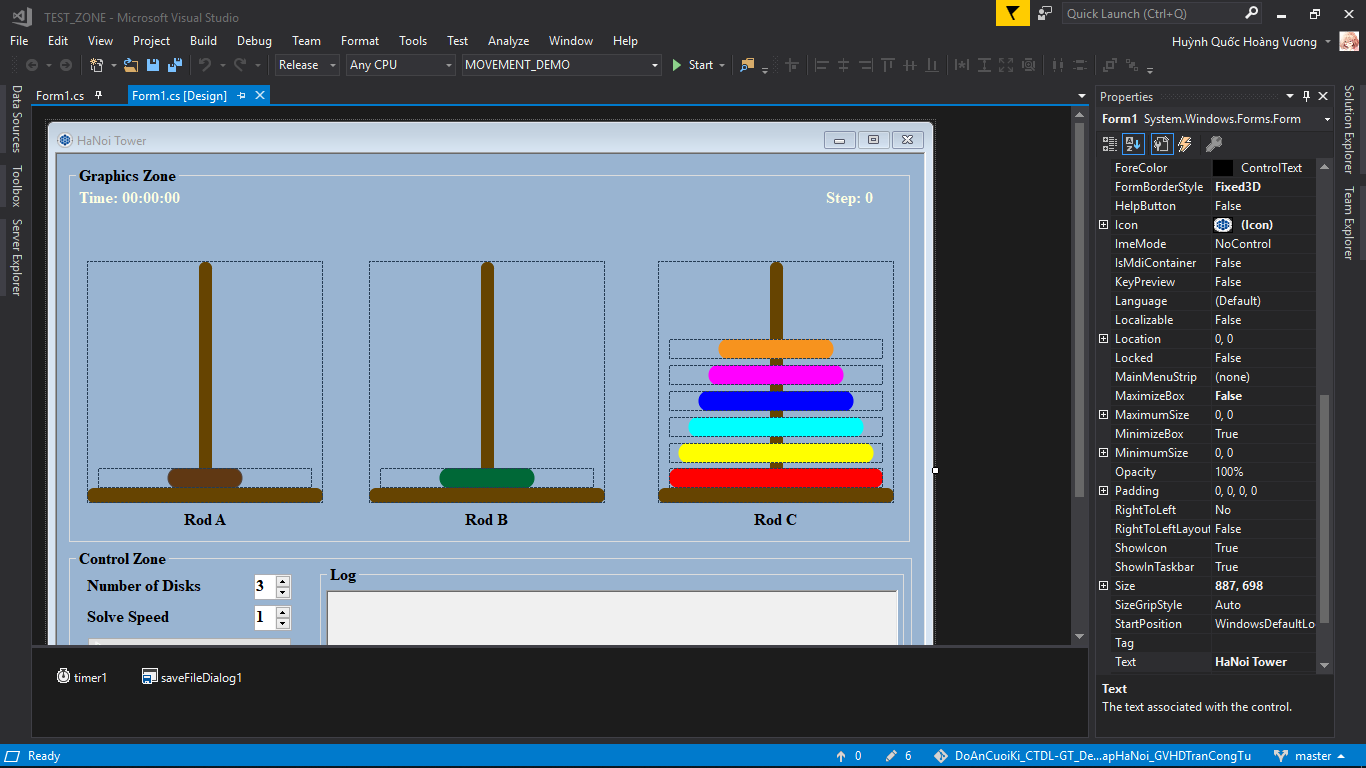
Các lần lặp tiếp tục chỉ xử lý việc chuyển 1 đĩa. Chương trình con in ra các phép chuyển và dẫn đến ngăn xếp rỗng.

* 1. *Quá trình và công việc thực hiện đồ án.*

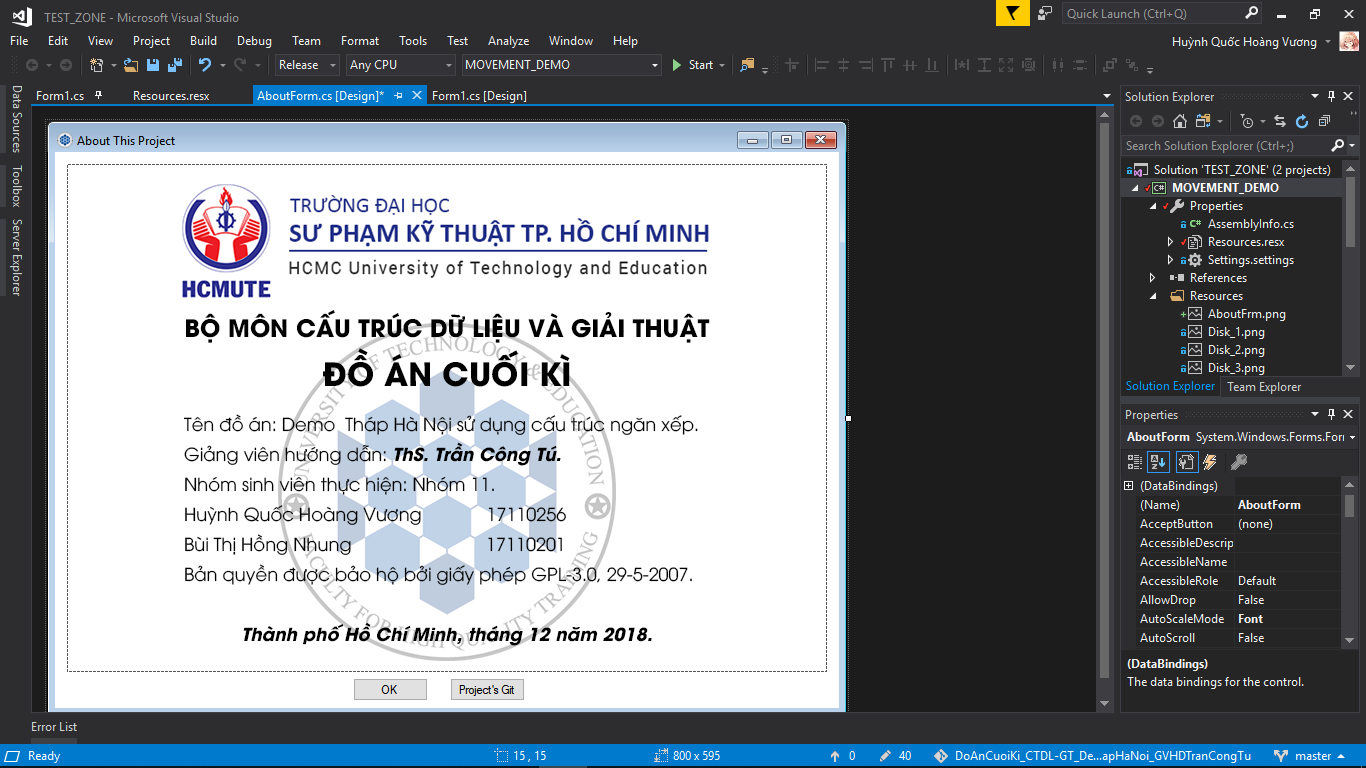
|  |  |  |
| --- | --- | --- |
| Tên công việc | Tên hàm | Người thực hiện |
|  |  |  |
|  |  |  |
|  |  |  |

(xoá cái này đi)

* 1. *Thiết kế giao diện, backdrop.*

**

*Hình 3: Form chính của chương trình.*

**

*Hình 4: Form giới thiệu.*

* 1. *Cài đặt thuật toán và viết chương trình.*
  2. *Kiểm lỗi và debug.*
  3. *Chương trình sản phẩm của đồ án.*
  4. *Cách thức hoạt động của chương trình.*

1. Kết luận và rút kinh nghiệm đồ án.
   1. *Bản kế hoạch hoàn thành.*
   2. *Ưu điểm.*
   3. *Nhược điểm.*
   4. *Phương án cải thiện.*
2. Phụ lục.
3. Tài liệu tham khảo.