

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO CUỐI KỲ: PHÂN TÍCH
VÀ THIẾT KẾ GIẢI THUẬT

QUY HOẠCH ĐỘNG 1 (DYNAMIC PROGRAMMING 1)

Người hướng dẫn: Ts Bùi Thanh Hùng

Người thực hiện: Trần Quốc Lĩnh - 51703124

Nguyễn Lê Minh Khang - 51703110

Khoá: 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

LỜI CẢM ƠN

Cảm ơn thầy Bùi Thanh Hùng đã dạy môn phân tích và thiết kế giải thuật hết sức nhiệt tình và tâm huyết, cảm ơn thầy đã định hướng và tạo cơ sở để chúng em có thể tự tìm hiểu và học thêm về các giải thuật.

Còn đây là đồ án của chúng em, một nội dung trong trường trình giảng dạy mà thầy đã giao cho chúng em. Trong quá trình làm bài đồ án này, chúng em vẫn còn nhiều thiếu sót. Chúng em rất mong nhận được sự đánh giá và chỉ bảo từ thầy!

BÀI TẬP LỚN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là sản phẩm bài tập lớn của riêng chúng tôi. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu, hình ảnh được chính chúng tôi thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài tiểu luận còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài tập lớn của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 23 tháng 11 năm 2019

Tác giả 1

(Đã ký)

Trần Quốc Lĩnh

Tác giả 2

(Đã ký)

Nguyễn Lê Minh Khang

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày 23 tháng 11 năm 2019
(*kí và ghi họ tên*)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày 23 tháng 11 năm 2019
(*kí và ghi họ tên*)

TÓM TẮT

Nội dung của bài báo cáo trình bày về đề tài quy hoạch động được thể hiện trong 4 chương. Trong đó có việc nêu lên các bài toán cơ bản, cách đặt vấn đề, cách tiếp cận, ý nghĩa. Đồng thời báo cáo cũng đưa ra hướng giải quyết cho các bài đó bằng các phương pháp cụ thể, chỉ ra được cách đánh giá cho các giải pháp đó. Bên cạnh đó thì nội dung của bài báo cáo còn có demo chi tiết đối với từng bài toán theo yêu cầu của đề bài, từ đó rút ra những đánh giá, kết luận cho toàn bộ bài báo cáo.

MỤC LỤC

LỜI CẢM ƠN	3
CAM KẾT	4
ĐÁNH GIÁ CỦA GIÁO VIÊN	5
TÓM TẮT	6
MỤC LỤC	7
DANH MỤC CHÚ THÍCH CÁC HÌNH ẢNH	8
CHƯƠNG 1: PHÁT BIỂU BÀI TOÁN	9
1.1. Đặt vấn đề	9
1.2. Cách tiếp cận	9
1.3. Ý nghĩa khoa học và thực tiễn của bài toán	9
CHƯƠNG 2: GIẢI QUYẾT BÀI TOÁN	10
2.1 Tổng quát về phương pháp giải quyết bài toán	10
2.2 Đặc trưng của giải pháp đề xuất	10
2.3 Cách đánh giá	10
CHƯƠNG 3: THỰC NGHIỆM	11
3.1. Dữ liệu	11
3.2. Công nghệ	11
3.2.1. Ngôn ngữ lập trình	11
3.2.2. Thư viện	11
3.3. Thực nghiệm	11
3.4. Đánh giá kết quả	11
CHƯƠNG 4: KẾT LUẬN	11
4.1 Kết quả đạt được	11
4.1.1 Kết quả	11
4.1.2. Hạn chế	11
4.2. Hướng phát triển	11
TÀI LIỆU THAM KHẢO	19

DANH MỤC CHÚ THÍCH CÁC HÌNH ẢNH

Hình ảnh

Hình 1: Nội dung hình 1

Hình 2: Nội dung hình 2

CHƯƠNG 1: PHÁT BIỂU BÀI TOÁN

1.1 Đặt vấn đề.

Bài tập cuối kỳ này thể hiện cách giải các bài toán theo phương pháp quy hoạch động qua ba bài toán: chuỗi con chung dài nhất (Longest Common String), chuỗi tăng dài nhất (Longest Increase Sequence) và cây nhị phân vô hạn.

Nội dung bài toán:

1. Dây con chung dài nhất LCS (Longest Common Subsequence): Nhập vào 2 xâu A và B, tìm xâu con chung dài nhất của 2 xâu đó, in ra xâu con chung.
2. Dây con tăng dài nhất LIS (Longest Increasing Susequence): Nhập vào 1 dãy số, hãy tìm dãy con tăng dài nhất của dãy số đó, in ra kết quả. (1.5 điểm/8)
3. Trong một cây nhị phân vô hạn:
 - Mỗi nút có đúng 2 con – một con trái và một con phải.
 - Nếu một nút được gán nhãn bằng số nguyên X , thì con trái của nó được gán nhãn $2 \times x$ và con phải của nó được gán nhãn $2 \times x + 1$.
 - Gốc của cây được gán nhãn 1.

Một cuộc dạo chơi trên cây nhị phân bắt đầu từ gốc. Tại mỗi bước, ta sẽ nhảy tới con trái hoặc con phải của nút hiện thời, hoặc là dừng lại tại chính nút đó để nghỉ.

Một cuộc dạo chơi được mô tả bằng một chuỗi các chữ cái 'L', 'R' và 'P':

- 'L' thể hiện bước nhảy tới con trái;
- 'R' thể hiện bước nhảy tới con phải;
- 'P' thể hiện việc dừng để nghỉ.

Giá trị của một cuộc dạo chơi là nhãn của nút mà chúng ta kết thúc. Ví dụ, giá trị của cuộc dạo chơi LR là 5, trong khi giá trị của cuộc dạo chơi RPP là 3.

Một tập hợp các cuộc dạo chơi được mô tả bởi một chuỗi các kí tự 'L', 'R', 'P' và '*'. Mỗi dấu '*' có thể là một trong 3 cách di chuyển; tập hợp các cuộc dạo chơi chứa tất cả các cuộc dạo chơi thích hợp với khuôn mẫu đó.

Ví dụ, tập hợp L^*R chứa các cuộc dạo chơi LLR, LRR và LPR. Tập hợp $**$ chứa các cuộc dạo chơi LL, LR, LP, RL, RR, RP, PL, PR và PP.

Cuối cùng, giá trị của một tập hợp các cuộc dạo chơi đúng bằng tổng các giá trị của tất cả các cuộc dạo chơi trong tập hợp đó.

Tính giá trị của một tập hợp các cuộc dạo chơi cho trước.

1.2 Cách tiếp cận.

Theo yêu cầu của đề tài, các bài toán trên phải được giải theo phương pháp quy hoạch động, một phương pháp giảm thời gian chạy của các thuật toán thể hiện các tính chất của các bài toán con gối nhau (overlapping subproblem) và cấu trúc con tối ưu (optimal substructure).

Quy hoạch động thường dùng một trong hai cách tiếp cận:

- top-down (Từ trên xuống): Bài toán được chia thành các bài toán con, các bài toán con này được giải và lời giải được ghi nhớ để phòng trường hợp cần dùng lại chúng.
- bottom-up (Từ dưới lên): Tất cả các bài toán con có thể cần đến đều được giải trước, sau đó được dùng để xây dựng lời giải cho các bài toán lớn hơn.

1.3 Ý nghĩa khoa học và thực tiễn của bài toán.

Ba bài toán trên có nhiều ý nghĩa khoa học và thực tiễn từ vi mô đến vĩ mô. Trong đó, những ý nghĩa quan trọng nhất là:

- Bài toán LCS hỗ trợ trong việc so sánh chuỗi và mức độ tương đồng giữa các chuỗi, các đoạn văn. Có thể ứng dụng vào việc xây dựng hệ thống trả lời câu hỏi tự động.
- Bài toán LIS giúp phát hiện chuỗi tăng dài nhất của một dãy số, qua đó có thể sử dụng cho các công việc liên quan đến sự tăng trưởng của một tiến trình nào đó.
- Giúp sinh viên nắm rõ hơn bản chất của các bài toán quy hoạch động và có thể sử dụng thành thạo các giải thuật, phương pháp chia nhỏ bài toán từ đó áp dụng vào các vấn đề lớn trong thực tế.

CHƯƠNG 2: GIẢI QUYẾT BÀI TOÁN

2.1 Tổng quát về phương pháp giải quyết bài toán.

Các bài toán sẽ được chia nhỏ ra thành các phần đơn giản, mà có thể dễ dàng giải quyết bằng các phương pháp thông thường. Ta dùng đệ quy hoặc không đệ quy kết hợp các biến lưu trữ để chia nhỏ bài toán.

2.2 Đặc trưng của giải pháp đề xuất.

Phương pháp giải quyết đã đề xuất ở trên mang tính tổng quát, áp dụng được cho hầu hết trường hợp. Dễ thực hiện, số câu lệnh ngắn.

2.3 Cách đánh giá.

Các tiêu chí để đánh giá gồm có:

- Tính chính xác của kết quả trả về.
- Độ phức tạp O của thuật toán.
- Mức độ khó dễ của việc cài đặt giải thuật.

CHƯƠNG 3: THỰC NGHIỆM

3.1 Dữ liệu.

Các testcase được tìm kiếm trên mạng hoặc do chính người lập trình nghĩ ra để test nhiều trường hợp khác nhau. Khi báo cáo có thể sẽ có thêm nhiều testcase khác do người hướng dẫn cung cấp.

3.2 Công nghệ.

3.2.1 Ngôn ngữ lập trình.

Ngôn ngữ lập trình được sử dụng trong bài báo cáo này là Python. Ngoài việc Python là ngôn ngữ bắt buộc khi học môn học này thì nó đang dần trở nên phổ biến hơn ngày nay. Sự phát triển của nhiều ngành như trí tuệ nhân tạo, học máy một cách nhanh chóng, các thuật toán ngày càng trở nên tinh vi hơn thì việc sử dụng Python sẽ giúp chúng ta tiếp cận mọi thứ một cách dễ dàng, hiệu quả hơn. Ngoài ra, Python còn rất dễ sử dụng, dễ cài đặt, vì thế nên nó càng trở nên dễ tiếp cận với mọi người.

3.2.2 Thư viện.

Bài báo cáo chỉ gồm 3 bài tập theo yêu cầu và đây lại là môn học phân tích và thiết kế giải thuật nên bài báo cáo không sử dụng thư viện nào hỗ trợ. Đây lại là những bài toán cơ bản, điển hình cho chủ đề của đề án nên việc sử dụng thư viện có thể không cần thiết, sinh viên có thể tự hoàn thiện code của mình mà không cần thư viện.

3.3 Thực nghiệm.

3.3.1 Bài toán dãy con tăng dài nhất.

Ý tưởng

- Tạo một mảng lưu giá trị của từng phần tử trong dãy số, có giá trị đầu là 0.
- Duyệt dãy số từ cuối ngược lên đầu. Tại mỗi phần tử, giá trị mới của nó bằng giá trị lớn nhất trong số các phần tử lớn hơn nó cộng thêm 1.
- Từ mảng các giá trị chọn ra dãy con tăng dài nhất của dãy số.

Mảng lưu giá trị	2	3	2	0	0	1	0
Dãy số	3	1	2	7	6	4	5

Hình 1: Mảng giá trị của dãy số.

Mã giả

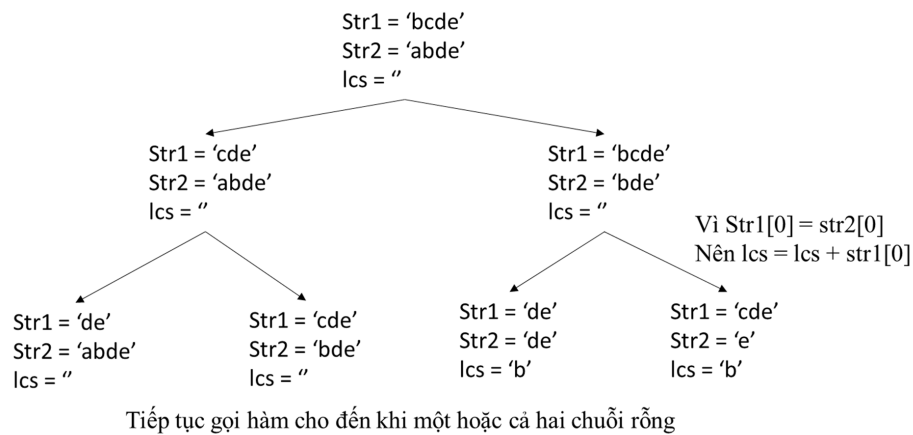
Algorithm 1: Chuỗi con tăng dài nhất

```
1 function LIS (array)
    Đầu vào: Mảng các số nguyên
    Đầu ra : Chuỗi con tăng dài nhất
2   val is an array of  $n$  zeros;
3   for  $i \leftarrow n - 1$  to 0 do
4       for  $j \leftarrow i$  to  $n - 1$  do
5           if  $array_j > array_i$  and  $val_j \geq val_i$  then
6                $val_i \leftarrow val_j + 1$ ;
7   lis is an interger array;
8    $m \leftarrow \max(val)$ ;
9   for  $i \leftarrow i$  to  $n - 1$  do
10      if  $val_i = m$  then
11          lis.Insert(array( $i$ ));
12           $m \leftarrow m - 1$ 
13  return lis
```

3.3.2 Bài toán dãy con chung dài nhất LCS.

Ý tưởng

- Hai chuỗi đầu vào sẽ được tách nhỏ ra để tìm kiếm dãy con chung dài nhất sau đó tổng hợp lại và chọn chuỗi dài nhất mà hàm tìm được.
- Bài toán sẽ được giải theo đệ quy với điều kiện dừng là khi một trong hai (hoặc cả hai) chuỗi không còn ký tự để so sánh.



Hình 2: Minh họa hàm đệ quy LCS.

Mã giả

Algorithm 2: Chuỗi con chung dài nhất

```

1 function LCS (str1, str2, lcs)
    Đầu vào: 2 chuỗi cần tìm dãy con chung và 1 biến lưu giá trị
    Đầu ra : Chuỗi con chung dài nhất
2   if str1 = 0 or str2 = 0 then
3       | return lcs;
4   if str1[0] = str2[0] then
5       | str1.Remove(str1[0]);
6       | str2.Remove(str2[0]);
7       | lcs ← lcs + str1[0];
8       | return LCS(str1, str2, lcs);
9   else
10      | newstr1 = str1;
11      | newstr1.remove(str1[0]);
12      | child1 = LCS(newstr1, str2, lcs);
13      | newstr2 = str2;
14      | newstr2.remove(str2[0]);
15      | child2 = LCS(str1, newstr2, lcs);
16      | return max(child1, child2, key=length);

```

3.3.3 Bài toán cây nhị phân vô hạn.

Ý tưởng

- Chuỗi lưu các node trên cây trong cuộc dạo chơi sẽ được cắt thành các cuộc dạo chơi ngắn hơn sau đó tổng hợp lại để tính giá trị cuối cùng.
- Dùng một biến lưu trữ giá trị hiện thời của node, với giá trị đầu là 1.

$$\begin{aligned}
 & \text{crr_val} = 1 \\
 & \text{crr_val} \times L * R = 1 \times (L * R) = (1 \times L) \times (* R) = 2 \times (* R) \\
 & \text{crr_val} = 2 \\
 & \text{crr_val} \times (* R) = 2 \times (* R) = [2 \times (L R)] + [2 \times (P R)] + [2 \times (R R)] \\
 & = \\
 & \begin{array}{lll}
 \text{crr_val} = 2 & \text{crr_val} = 2 & \text{crr_val} = 2 \\
 2 \times (P R) = (2 \times P) R = 2 \times R & 2 \times (L R) = (2 \times L) R = 4 \times R & 2 \times (R R) = (2 \times R) R = 5 \times R \\
 \text{crr_val} = 2 & \text{crr_val} = 4 & \text{crr_val} = 5 \\
 \text{crr_val} \times R = 2 \times R = 4 & \text{crr_val} \times R = 4 \times R = 9 & \text{crr_val} \times R = 5 \times R = 11 \\
 & = 25
 \end{array}
 \end{aligned}$$

Hình 3: Minh họa cách tính giá trị của chuyển đi trên cây.

Mã giả

Algorithm 3: Cây nhị phân vô hạn

```
1 function inf_tree(str, crr_val)
    Đầu vào: 1 chuỗi lưu các node và 1 biến lưu giá trị
    Đầu ra : Giá trị cuối cùng của chuyển đạo chơi
2 if str.isEmpty() then
3     | return crr_val;
4 crr_pos  $\leftarrow$  str[0];
5 str.Remove(str[0])
6 if crr_pos = 'P' then
7     | return inf_tree(str, crr_val);
8 else if crr_pos = 'L' then
9     | return inf_tree(str, crr_val  $\times$  2);
10 else if crr_pos = 'R' then
11     | return inf_tree(str, crr_val  $\times$  2 + 1);
12 else if crr_pos = '*' then
13     | P  $\leftarrow$  inf_tree(str, crr_val);
14     | L  $\leftarrow$  inf_tree(str, crr_val  $\times$  2);
15     | R  $\leftarrow$  inf_tree(str, crr_val  $\times$  2 + 1);
16     | return P + L + R;
```

3.4 Đánh giá kết quả.

Từ những thực nghiệm đã được nêu như trên có thể rút ra đánh giá cho bài báo cáo là cơ bản hoàn thành các yêu cầu của đề án. Bài báo cáo đã test qua nhiều testcase khác nhau và chính xác ở mọi trường hợp được test. Phương pháp quy hoạch động cũng được chỉ ra ở phần code trên. Vì vậy, đây có thể coi là kết quả tương đối tốt cho bài báo cáo này, hoàn thành không chỉ yêu cầu về mặt kết quả mà còn về mặt thuật toán.

Chương 4: KẾT LUẬN

4.1 Kết quả đạt được.

4.1.1 Kết quả.

Qua những điều được nêu lên ở các chương phía trên, ta có thể thấy bài báo cáo đã trình bày một cách cơ bản nhất chủ đề của đề án là quy hoạch động, thông qua ba bài toán được giao. Mỗi bài toán đều được nêu lên nội dung, chỉ ra cách tiếp cận chung cho bài toán dạng quy hoạch động và cũng nói lên được ý nghĩa khoa học của nó không chỉ trong nghiên cứu mà còn có thể áp dụng vào những việc thực tiễn của cuộc sống.

Bài báo cáo còn nêu lên phương hướng giải quyết một cách tổng quát nhất, đặc trưng của phương hướng đó, đồng thời là các cách đánh giá gồm nhiều tiêu chí, để từ đó ta có thể thấy độ hiệu quả của giải thuật được nêu.

Phần thực nghiệm cũng cho thấy những kết quả đáng mong đợi từ việc áp dụng những phương pháp đã nêu vào giải quyết các vấn đề được giao. Đó việc đưa ra được mã giả kết hợp cùng việc giải thích, code demo đã cho ra nhiều kết quả theo đúng kì vọng mà những testcase của người lập trình đã đề ra. Qua đó, ta có thể kết luận một lần nữa là bài báo cáo cơ bản đã hoàn thành yêu cầu đề ra.

4.1.2 Hạn chế.

Bài báo cáo tuy đã hoàn thiện cơ bản nhất có thể nhưng vẫn còn tồn tại một vài thiếu sót nhỏ. Các phần theo yêu cầu đã được hoàn thành nhưng còn ở mức là tổng quát, chung nhất, chưa đi vào một cách cụ thể, có thể do phần quy hoạch động còn ít tài liệu tham khảo. Phương pháp giải quyết cũng chính là đề tài được nêu nên rất khó cho việc sáng tạo hay đổi mới. Vì vậy, bài báo cáo nếu muốn nâng cao hơn cần tìm hiểu rất nhiều mới có thể cải tiến và phát triển hơn được.

4.2 Hướng phát triển.

Có nhiều hướng phát triển khác nhau cho các bài toán dạng quy hoạch động. Ngoài việc xây dựng thuật toán theo kiểu bài báo cáo đã nêu thì còn có nhiều thuật toán tối ưu khác giúp cho việc nâng cao hơn ứng dụng sau này. Ngành công nghệ thông tin ngày càng phát triển, bên trong đó là các lĩnh vực trí tuệ nhân tạo, học máy cũng ứng dụng việc sử dụng quy hoạch động để tạo nên cách thuật toán tìm kiếm, hay build một con AI chatbox,... và còn nhiều ứng dụng khác nữa. Có thể xây dựng cho các chương trình một giao diện người dùng để có thể dễ tương tác hơn cho người dùng cuối.

TÀI LIỆU THAM KHẢO

Tiếng Việt

Wikipedia > Quy hoạch động