

BIỂU DIỄN TRỰC QUAN VÀ TỔ CHỨC GIẢI BÀI TOÁN THEO BA GIAI ĐOẠN TRONG DẠY HỌC THUẬT TOÁN CHO SINH VIÊN CAO ĐẲNG TIN HỌC

LÊ THỊ BẢO NGỌC*

Ngày nhận bài: 10/06/2016; ngày sửa chữa: 15/06/2016; ngày duyệt đăng: 20/06/2016.

Abstract: The algorithms are very difficult for teachers to teach. This requires innovation of teaching methods. This article refers to application of case study and visual performances in teaching algorithm for students of Computing college and proposes measures to improve quality of teaching.

Keywords: Case study, teaching method, visual performance.

Một trong những thách thức của giảng viên (GV) ở các chuyên ngành Tin học là những khó khăn khi dạy học các thuật toán. Thông thường, kiến thức về thuật toán có trong các chuyên ngành liên quan đến lập trình, sinh viên (SV) được giới thiệu một số ngôn ngữ lập trình để mô tả các thuật toán đơn giản. Do vậy, để hướng tới việc giảng dạy thuật toán dễ hiểu và tường minh, theo chúng tôi, GV cần sử dụng phương pháp dạy học theo tình huống kết hợp với phương pháp dạy học trực quan để phát huy được hiệu quả tối đa trong giảng dạy thuật toán cho SV ngành Tin học.

1. Thuật toán và phân tích, đánh giá thuật toán

Thuật toán là một dãy hữu hạn các bước, mỗi bước mô tả chính xác các phép toán hoặc hành động cần thực hiện để giải quyết vấn đề.

Việc thiết kế các thuật toán hiệu quả là một trong những lĩnh vực nghiên cứu của ngành khoa học máy tính. Khi chọn lựa thuật toán, cần dựa trên cơ sở phân tích thuật toán và đánh giá độ phức tạp để chọn được thuật toán tốt nhất. SV lần đầu tiên làm quen với thuật toán sẽ gặp nhiều khó khăn về các bước thực hiện, hiệu quả cũng như khả năng thực thi của thuật toán. Vì vậy, cần sử dụng nhiều phương pháp tiếp cận khác nhau để nâng cao hiệu quả dạy học, giúp SV hiểu sâu sắc và toàn diện về thuật toán.

2. Dạy học thuật toán cho SV ngành Tin học

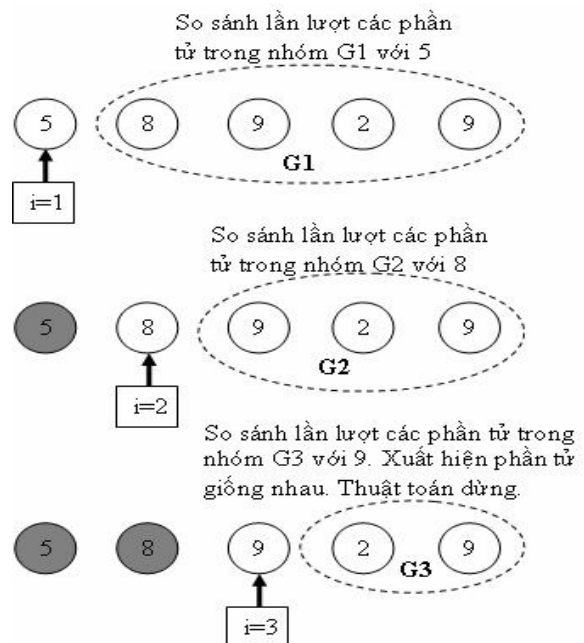
Trong quá trình dạy học thuật toán, ngoài việc cung cấp cho SV một số thuật toán, GV cần kết hợp phân tích, đánh giá thuật toán để các em thấy được tính hiệu quả của thuật toán. Thông thường, SV có khả năng phân tích và đánh giá thời gian thực hiện các thuật toán nhưng đa số các em gặp khó khăn trong việc thiết kế thuật toán. Để khắc phục tình trạng này, GV nên sử dụng các bài tập được thiết kế theo 03 giai đoạn sau:

Giai đoạn 1. Giải bài toán cụ thể bằng thuật toán đơn giản.

Trong quá trình dạy học thuật toán, GV cần đưa ra tình huống ở một bài toán cụ thể. Thông qua bài toán đó, GV mô tả các bước giải quyết vấn đề bằng hình ảnh trực quan kết hợp với sự giải thích để SV có thể hiểu rõ thuật toán.

Chẳng hạn, với bài toán: cho một mảng bất kì, kiểm tra xem tất cả các phần tử trong mảng có khác nhau không? Thuật toán sẽ dừng khi có phần tử trong mảng giống nhau hoặc sau khi duyệt hết mảng nhưng vẫn chưa tìm thấy phần tử giống nhau. GV tiến hành mô tả cách giải quyết vấn đề bằng hình ảnh trực quan cho cả hai trường hợp thông qua mảng có giá trị như sau:

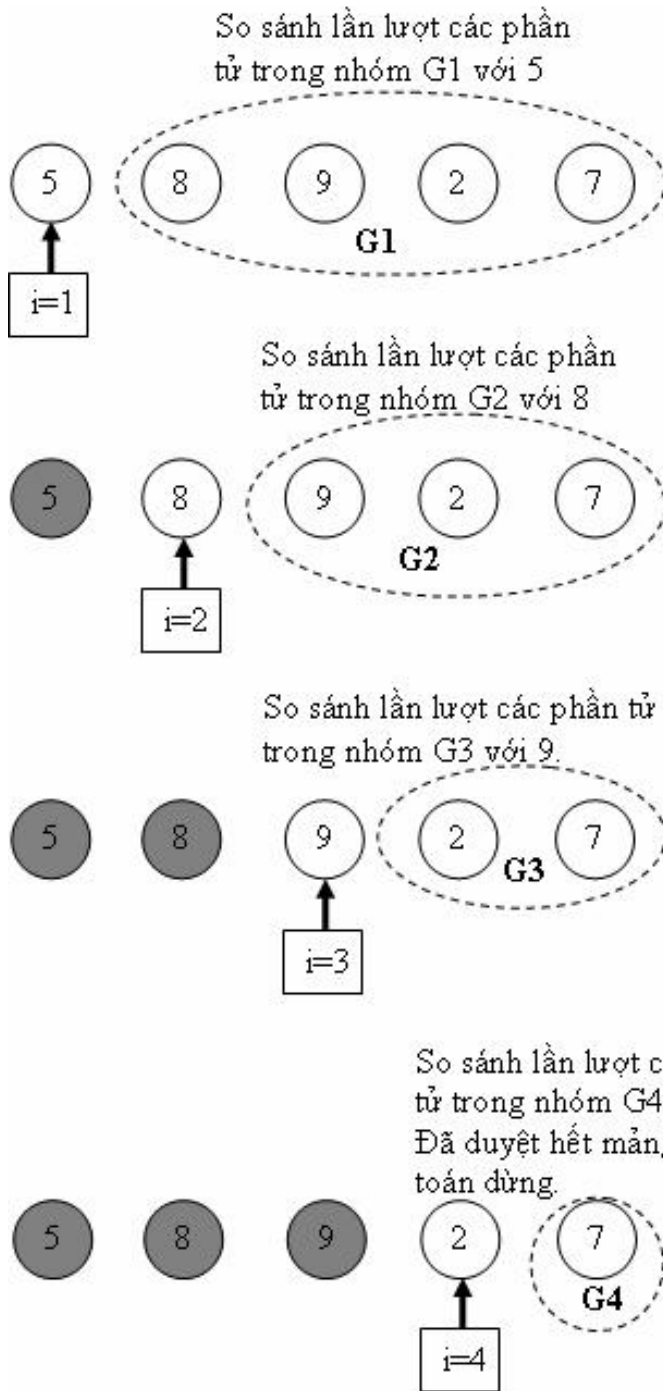
- Trường hợp 1: Mảng chứa phần tử giống nhau.



* Trường Cao đẳng Sư phạm Kiên Giang

- Trường hợp 2: Mảng không chứa các phần tử giống nhau. tiếp thu của SV.

Giai đoạn 2. Trình bày thuật toán bằng ngôn ngữ lập trình.



Sau giai đoạn 1, GV tiến hành trình bày thuật toán. Quá trình này cần thể hiện được cụ thể từng công việc ở mỗi bước, dữ liệu vào và dữ liệu ra. GV có thể dùng ngôn ngữ thông thường và kí hiệu toán học để trình bày thuật toán. Tuy nhiên, mỗi bước của thuật toán cần được mô tả chính xác, chỉ có một cách hiểu duy nhất. Ngoài việc biểu diễn bằng ngôn ngữ thông thường, thuật toán cần được mô tả bằng các ngôn ngữ lập trình. Bởi trong các ngôn ngữ này, các mệnh lệnh được tạo thành theo quy tắc cú pháp nghiêm ngặt và chỉ có một ý nghĩa duy nhất. Chẳng hạn, bài toán “kiểm tra xem các phần tử trong mảng có trùng nhau không?” được viết bằng ngôn ngữ C++ như sau:

```
int test (int a[n])
{int i;
for(i=0; i<n-1; i++)
for(j=i+1; j<n; j++)
if(a[i]==a[j])
return0;
return1;}
```

Giai đoạn 3. Phân tích, tính độ phức tạp của thuật toán và đặt ra tình huống muốn cải thiện thuật toán ban đầu. Ở giai đoạn này, GV chỉ ra những hạn chế của thuật toán vừa trình bày. Điều này sẽ giúp SV hiểu được tầm quan trọng của việc phân tích, đánh giá thuật toán và nhận ra rằng có một số thuật toán không hợp lí.

GV giải thích tầm quan trọng của việc giảm thiểu tối đa thời gian chạy của thuật toán và tính độ phức tạp của thuật toán. Sau đó đưa ra câu hỏi tình huống, nhằm

Với cách diễn đạt trực quan như vậy, SV có thể hiểu thuật toán dễ dàng hơn. Quá trình mô tả bằng hình ảnh trực quan sẽ hiệu quả hơn nếu biết kết hợp nhiều cách khác nhau thay vì dùng một cách. Sử dụng tổ hợp nhiều cách giúp GV tạo ra các mô hình khác nhau; qua đó, tăng cường sự hiểu biết và khả năng

cải thiện thuật toán vừa trình bày, GV đóng vai trò là người hướng dẫn. Thông qua những câu hỏi tình huống, SV từng bước thiết kế thuật toán mới. Bằng cách này, thay vì tiếp nhận tri thức một cách thụ động, SV tích cực tham gia vào quá trình hình thành tri thức mới. Nếu SV nào đó có hướng giải quyết không

phù hợp, còn thiếu sót sẽ được các SV khác hoặc GV hỗ trợ trong quá trình giải quyết vấn đề.

Với bài toán “kiểm tra các phần tử trong mảng có trùng nhau không?”, GV có thể đưa ra nhận xét độ phức tạp $O(n^2)$ là khá lớn. Vậy, có thuật toán nào giải được bài toán “kiểm tra các phần tử của mảng có trùng nhau không?” với độ phức tạp nhỏ hơn $O(n^2)$? Giả sử ta thực hiện việc sắp xếp trước khi kiểm tra các phần tử trong mảng thì thuật toán có độ phức tạp là bao nhiêu? Rõ ràng, thuật toán sắp xếp tốt có độ phức tạp là $O(n \log n)$, thuật toán kiểm tra tuần tự sau khi sắp xếp chỉ có độ phức tạp $O(n)$. Do đó, toàn bộ thuật toán “kiểm tra các phần tử trong mảng có trùng nhau không?” chỉ có độ phức tạp là $O(n \log n)$, nhỏ hơn độ phức tạp $O(n^2)$. Từ đó, HS dựa trên những gợi ý để thiết kế một thuật toán mới.

3. Một số bài tập được thiết kế theo ba giai đoạn

Ví dụ 1: Cho mảng a có chiều dài n. Tính độ sai biệt lớn nhất giữa 2 phần tử trong mảng.

GV tổ chức cho SV giải bài toán theo quy trình ba giai đoạn sẽ được kết quả như sau:

Giai đoạn 1: Mô tả trực quan thuật toán ở mức đơn giản để giải quyết bài toán.

Giai đoạn 2: Trình bày thuật toán bằng ngôn ngữ lập trình.

```
int difference (int a[n])
{int tam = 0, i, j;
  for (i = 0; i < n-1; i++)
    for (j = i+1; j < n; j++)
      if (abs(a[j] - a[i]) > tam)
        tam = abs(a[j] - a[i]);
  return tam;}
```

Giai đoạn 3: Thuật toán có độ phức tạp $O(n^2)$. Giả sử, thực hiện tìm phần tử lớn nhất và nhỏ nhất trong mảng, sau đó tính độ sai biệt giữa hai phần tử này thì bài toán sẽ được giải quyết với thuật toán có độ phức tạp là $O(n)$, nhỏ hơn độ phức tạp của thuật toán ban đầu.

Ví dụ 2: Cho mảng a, b với chiều dài n, a được sắp xếp theo thứ tự tăng dần và b chưa được sắp xếp. Kiểm tra xem có tồn tại một phần tử trong mảng b có giá trị bằng tổng hai phần tử liên tiếp trong mảng a không?

GV tổ chức cho SV giải bài toán theo quy trình ba giai đoạn sẽ được kết quả như sau:

Giai đoạn 1: Mô tả trực quan thuật toán ở mức đơn giản để giải quyết bài toán.

Giai đoạn 2: Trình bày thuật toán bằng ngôn ngữ lập trình

```
int test (int a[n], int b[n], int &i, int &j)
{for (j = 0; j < n; j++)
  for (i = 0; i < n-1; i++)
    if (b[j] == a[i] + a[i+1])
      return 1;
  return 0;}
```

Giai đoạn 3: Độ phức tạp của thuật toán là $O(n^2)$. Nếu đối với mỗi phần tử trong mảng b, sử dụng thuật toán tìm kiếm nhị phân tìm hai phần tử liên tiếp trong mảng a có tổng bằng phần tử đó thì thuật toán có độ phức tạp là $O(n \log n)$, nhỏ hơn độ phức tạp $O(n^2)$.

Dạng bài tập được phân hóa theo ba giai đoạn ở trên là rất ít trong các tài liệu giảng dạy. Ở hầu hết các tài liệu, thuật toán tốt nhất được đưa ra cùng với độ phức tạp tương ứng của thuật toán. SV được tiếp xúc và hiểu rõ các thuật toán ở mức độ đơn giản, thô sơ nhất, từng bước hình thành thuật toán hiệu quả dựa trên quá trình giải quyết vấn đề của thuật toán ban đầu. Bằng cách này, SV nắm rõ những quy ước về một thuật toán hiệu quả cũng như biết cách tư duy khi tiến hành thiết kế một thuật toán nào đó. Từ đó, tạo hứng thú, sự say mê học tập cho SV khi học về thuật toán. □

Tài liệu tham khảo

- [1] Đỗ Xuân Lôi (2015). *Giáo trình cấu trúc dữ liệu và giải thuật*. NXB Giáo dục.
- [2] Dương Tử Cường (2005). *Ngôn ngữ lập trình C++*. NXB Khoa học kỹ thuật.
- [3] Trần Thông Quế (2014). *Cấu trúc dữ liệu và thuật toán*. NXB Thông tin và truyền thông.
- [4] Hồ Thuận (chủ biên) - Hồ Cẩm Hà - Trần Thiên Thành (2008). *Cấu trúc dữ liệu và phân tích thuật toán*. NXB Giáo dục.
- [5] Phan Trọng Ngọ (2005). *Phương pháp dạy học trong nhà trường*. NXB Đại học Sư phạm.

THÔNG BÁO

Căn cứ Kế hoạch phát triển Tạp chí Giáo dục giai đoạn 2016-2020, căn cứ công văn số 37/HĐCDGSNN ngày 4/4/2016 của Hội đồng Chức danh Giáo sư Nhà nước, từ tháng 6/2016, Tạp chí Giáo dục sẽ **không ghi học hàm, học vị của tác giả trong bài báo khoa học**.

TẠP CHÍ GIÁO DỤC