Câu hỏi ôn tập

Phần 1

1. Nêu các thành phần cơ bản của một chương trình và cho biết mối quan hệ của các thành phần đó

- Các thành phần cơ bản :

+ Cấu trúc dữ liệu

+ Giải thuật

- Mqh:Có mqh chặt chẽ vs nhau

+ Đối tượng xử lí của giải thuật là dữ liệu

+ Dữ liệu chứa các thông tin cần thiết để thực hiện giải thuật

1. Thành phần cấu trúc dữ liệu có tác động đến những yếu tố nào của chương trình

* Các cấu trúc dữ liệu cơ bản hay dùng là: *mảng*(array), *danh sách*(list), *ngăn xếp*(stack), *hàng đợi*(queue),*cây*(tree), đồ thị (graph),..
* Lựa chọn giải thuật sao cho phù hợp
* Khó khăn trong việc triển khai chương trình
* Chất lượng và kết quả cuối cùng

1. Thành phần giải thuật có tác động đến yếu tố nào của chương trình

* Giải thuật là một hệ thống các thao  
  tác, các phép toán được thực hiện theo trình tự  
  nhất định trên một số đối tượng dữ liệu nào đó,  
  sao cho sau một số bước hữu hạn ta có được  
  kết quả mong muốn
* Thời gian cần thiết để hoàn thành chương trình

1. Theo bạn khi xây dựng chương trình, thành phần nào của chương trình nên được thiết kế trước? vì sao

- Giải thuật trc

- Vì :

+ phân tích cấu trúc dữ liệu nhằm hiểu chính xác vấn đề của bài

+ thiết lập mqh giữa dữ liệu và kết quả phải tìm

+ chọn đc giải thuật phù hợp nhất để giải quyết vấn đề bài toán

1. Độ phức tạp có thể cài đặt chương trình trên máy tính và độ phức tạp không thể cài đặt chương trình mà nên thiết kế lại.

* Một giải thuật mà thời gian thực hiện có độ phức tạp là một hàm đa thức thì chấp nhận được tức là có thể cài đặt để thực hiện, còn các giải thuật có độ phức tạp hàm mũ thì phải tìm cách cải tiến giải thuật.

1. Đệ quy là gì, khi nào giải thuật nên được thiết kế theo tiếp cận đệ quy

* Một đối tượng gọi là đệ qui khi nó được định nghĩa qua chính nó.
* Giải thuật nên được thiết kế theo tiếp cận đệ quy khi:
  + Định nghĩa số tự nhiên
  + Định nghĩa xâu ký tự
  + Định nghĩa hàm giai thừa

1. Trình bày các phương pháp đệ quy và độ phức tạp tương ứng

* Đệ quy tuyến tính, độ phức tạp: O(n)
* Đệ quy nhị phân, độ phức tạp: O(n)
* Đệ quy phi tuyến, Độ phức tạp: O(n)^n
* Đệ quy tương hỗ, độ phức tạp: O(n)\*G(m)

1. Đệ quy tuyến tính là gì, khi nào ta sử dụng đệ quy tuyến tính

* Trong thân hàm có duy nhất một lời gọi hàm gọi lại chính nó một cách tường minh.
* Vd: sử dụng đệ quy tuyến tính để tính hàm giai thừa

1. Đệ quy nhị phân là gì, khi nào ta sử dụng đệ quy nhị phân

* Đệ quy nhị phân là loại đệ quy trong thân hàm có hai lời gọi lại chính nó một cách tường minh
* Vd: dùng đệ quy nhị phân để tìm vị trí của n trong dãy fibonacii

1. Khi nào ta cần khử đệ quy cho chương trình

* Khi chương trình có thể viết dưới dạng lặp hoặc các cấu trúc lệnh khác thì không nên sử dụng đệ quy. Có 2 lí do chính:  
  + Khi một thủ tục đệ quy gọi chính nó , tập các đối tượng được sử dụng trong thủ tục này như các biến, hằng cấu trúc, .. sẽ được tạo ra
  + Sử dụng đệ quy đôi khi tạo ra những tính toán dư thừa

1. Tìm kiếm là gì, có nhưng phương pháp tìm kiếm nào và độ phức tạp của mỗi phương pháp

- Tìm kiếm là việc thu thập 1 số thông tin nào đó từ 1 khối thông tin đã đc lưu trữ trc đó

- Có 2 pp tìm kiếm :

+ Tìm kiếm tuyến tính (tìm kiếm tuần tự ) : độ phức tạp O(n)

+ Tìm kiếm nhị phân : độ phức tạp O(log(n))

1. Tìm kiếm tuyến tính (TKTT) là gì, khi nào ta sử dụng phương pháp TKTT. Phân tích ưu và nhược điểm của TKTT

- Tìm kiếm tuyến tính là phương pháp tìm kiếm 1 phần tử cho trước trong một danh sách bằng cách duyệt lần lượt từng phần tử của danh sách đó cho đến khi tìm thấy được phần tử mong muốn hoặc là đã duyệt qua hết danh sách

- Sử dụng phương pháp tìm kiếm tuyến tính khi dữ liệu chưa có sự sắp xếp

- Ưu nhược điểm :

+ Ưu : Thuật toán đơn giản dễ thực hiện, phù hợp để xử lí các bài toán nhỏ với dữ liệu chưa được sắp xếp

+ Nhược : Không phù hợp với các bài toán lớn hoặc tìm kiếm nhiều lần

1. Tìm kiếm nhị phân (TKNP) là gì, khi nào ta sử dụng phương pháp TKNP. Phân tích ưu và nhược điểm của TKNP

- Tìm kiếm nhị phân là : là thuật toán tìm kiếm một mảng đã sắp xếp bằng cách liên tục chia các khoảng tìm kiếm thành 1 nửa Bắt đầu với một khoảng từ phần tử đầu mảng, tới cuối mảng. Nếu giá trị của phần tử cần tìm nhỏ hơn giá trị của phần từ nằm ở giữa khoảng thì thu hẹp phạm vi tìm kiếm từ đầu mảng tới giửa mảng và nguợc lại. Cứ thế tiếp tục chia phạm vi thành các nửa cho dến khi tìm thấy hoặc đã duyệt hết.

- Sử dụng khi dữ liệu cần xử lí đã được sắp xếp theo 1 thứ tự nhất định

- Ưu nhược điểm

+ Ưu : Dễ dàng xử lí đối với những dữ liệu lớn ,có hiệu suất tốt

+ Nhược : Độ phức tạp lớn ,chỉ sử dụng được trên những mảng đã được sắp xếp

1. Sắp xếp là gì, có những phương pháp sắp xếp cơ bản nào và độ phức tạp tương ứng của mỗi phương pháp

- Sắp xếp là quá trình bố trí lại cá phần tử của một tập hợp theo thứ tự nào đó

- Những pp sắp xếp cơ bản :

+ sx kiểu lựa chọn (selection sort) : Độ phức tạp O(n2)

+ sx kiểu thêm dần (insertion sort) : Độ phức tạp O(n2)

+ sx kiểu nổi bọt (bubble sort): Độ phức tạp O(n2) (th tồi nhất)

+ sx nhanh (quick sort) : Độ phức tạp :

\* TH tốt nhất O(nlogn)

\* TH xấu nhất O(n2)

\* TH trung bình O(nlogn)

1. Phân tích sự giống và khác nhau giữa hai thuật toán selection sort và bubble sort

|  |  |  |
| --- | --- | --- |
|  | Selection sort | Bubble sort |
| Giống | - Đều là thuật toán sắp xếp  - Sử dụng phép toán hoán đổi  - Độ phức tạp O(n2) | |
| Khác | So sánh các phần tử để đặt phần tử nhỏ nhất vào vị trí phía trước | So sánh các phần tử để đặt phần tử nhỏ nhất vào vị trí phía sau |

1. Nêu cấu trúc dữ liệu được sử dụng cho thuật toán insert sort và đánh giá độ phức tạp của thuật toán trong ba trường hợp phân bố của dữ liệu

- Cấu trúc dữ liệu : dãy các đối tượng (các số ) trong 1 mảng

- Độ phức tạp

+tốt nhất : O(n)

+trung bình : O(n2)

+Tệ nhất : O(n2)

1. Nêu cấu trúc dữ liệu được sử dụng cho thuật toán Quick sort, phương pháp thiết kế của thuật toán?, phần tử chốt là gì và được dùng để làm gì.

Có những phương pháp chọn phần tử chốt là gì. Phân tích độ phức tạp của thuật toán do bị ảnh hưởng bởi việc chọn phần tử chốt.

* - Thuật toán sắp xếp Quick – Sort được thực hiện theo mô hình chia để trị.Thuật toán được thực hiện xung quanh một phần tử gọi là chốt (key).Mỗi phần tử chốt ở vị trí khác nhau sẽ cho ta một phiên bản khác nhau của thuật toán.
* - Phương pháp chọn chốt:
* + Luôn lựa chọn phần tử đầu tiên trong dãy làm chốt
* + Luôn lựa chọn phần tử cuối cùng trong dãy làm chốt
* + Luôn lựa chọn phần tử ở giữa dãy làm chốt
* + Lựa chọn phần tử ngẫu nhiên trong dãy làm chốt
* - Mỗi lần phân hoạch ta chọn phải phần tử có giá trị cực đại hoặc cực tiểu làm mốc.Khi đó dãy bị phân hoạch thành 2 phần không đều: 1 phần chỉ có một phần tử,phần còn lại có n-1 phần tử.Do đó, ta cần tới n lần phân hoạch mới xếp xong. Độ phức tạp O(n2)
* Trường hợp tốt nhất,trung bình. Độ phức tạp O(nlogn)

1. Nêu cấu trúc dữ liệu được sử dụng cho thuật toán Heap sort, phương pháp thiết kế của thuật toán?, quá trình sắp xếp phụ thuộc vào chức năng nào của đống, độ phức tạp trung bình của thuật toán là bao nhiêu.

**-** Thực hiện sắp xếp thông qua việc tạo các heap,trong đó heap là 1 cây nhị phân hoàn chỉnh có tính chất là khóa ở nút cha bao giờ cũng lớn hơn khóa ở các nút con.

Việc thực hiện giải thuật này được chia làm 2 giai đoạn

+ Đầu tiên là việc tạo heap từ dãy ban đầu.Theo định nghĩa của heap thì nút cha bao giờ cũng lớn hơn các nút con.Do vậy,nút gốc của heap bao giờ cũng là phần tử lớn nhất

+ Giai đoạn 2: Sắp xếp dựa trên heap tạo được. Phần tử bé nhất của dãy và được đặt lên đầu

* Một cây nhị phân hoàn toàn chỉnh có n nút thì chiều cao của cây đó là [log2(n+1)]. Khi tạo đống cũng như khi vun đống trong giai đoạn sắp xếp, trường hợp xấu nhất thì số lượng phép so sánh cũng chỉ tỷ lệ với chiều cao của cây
* Do đó có thể suy ra, trong trường hợp xấu nhất, cấp độ lớn của thời gian thực hiện Heap Sort là O(nlog2n).Việc đánh giá thời gian thực hiện trung bình giải thuật Heap Sort cũng là O(nlog2n)