



KỲ THI CUỐI KỲ – HỌC KỲ 1 / 2018-2019

Môn thi: Cấu trúc dữ liệu & giải thuật [CO2003]
GV ra đề: Trần Giang Sơn, Nguyễn Đức Dũng
Mã đề: 1814

Thời gian: 120'
Ngày thi: 25-12-2018

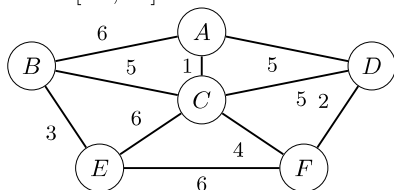
Sinh viên được mang tài liệu giấy. Sinh viên làm trực tiếp trên đề và nộp kèm bài làm phần tự luận.

I. Câu hỏi trắc nghiệm (3.0): sinh viên khoanh tròn vào câu trả lời đúng nhất

Ghi chú (với lớp TNMT): mỗi câu sai trong phần I, II bị trừ 0.1.

1. Tính ổn định của giải thuật sắp xếp: [6.4, 17]
 - (a) Không cho phép 2 phần tử dữ liệu có khóa giống nhau
 - (b) Cho phép 2 phần tử dữ liệu có khóa giống nhau
 - (c) Độ phức tạp thời gian trong ba trường hợp tốt, xấu và trung bình như nhau
 - (d) Cả 3 đáp án trên đều sai.
2. Biết rằng duyệt tiền thứ tự một cây nhị phân cho kết quả là A,B,C,D,E,K,F,G, kết quả duyệt trung thứ tự là B,C,A,D,K,E,F,G. Hãy cho biết kết quả duyệt hậu thứ tự [3.5, 15]
 - (a) C, B, K, G, F, E, D, A
 - (b) C, B, K, G, E, F, D, A
 - (c) C, B, G, K, F, E, D, A
 - (d) 3 đáp án trên đều sai
3. Xây dựng bảng hash từ dãy các từ khóa {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}, với hàm hash $H(key) = key \% 13$, sử dụng phương pháp danh sách liên kết (Linked list resolution) để giải quyết xung đột. Hãy cho biết có bao nhiêu phần tử trong danh sách liên kết ở địa chỉ 1. [5.3, 20]
 - (a) 3
 - (b) 4
 - (c) 5
 - (d) Tất cả đều sai
4. Giả sử biểu diễn đồ thị có trọng số G bằng ma trận kề M [7.1, 23]
 - (a) Nếu M đối xứng qua đường chéo chính, thì G là đồ thị vô hướng, G không thể là đồ thị có hướng
 - (b) Nếu M không đối xứng qua đường chéo chính, thì G là có thể là đồ thị vô hướng
 - (c) Số cạnh của đồ thị chính là số lượng giá trị khác 0 trong ma trận G
 - (d) Cả A, B và C đều sai
5. Giả sử bảng hash có không gian địa chỉ là $S[0..13]$, hàm $H(key) = key \% 11$, sử dụng phương pháp thăm dò bình phương (Quadratic probing) để giải quyết xung đột, trong bảng hash hiện đã có 4 khóa sau: 15, 38, 61, 84. Bây giờ chèn thêm khóa có giá trị 49 vào bảng hash, hãy cho biết khóa này sẽ được chèn vào địa chỉ nào. [5.2, 19]
 - (a) 3
 - (b) 5
 - (c) 8
 - (d) 9
6. Cho hàm đệ quy sau:
`void foo(int n) { if(n > 0) foo(n-1); }`
hãy cho biết nhận định nào sau đây đúng: [8.1, 11]
 - (a) Hàm đệ quy trên không có điểm dừng
 - (b) Hàm đệ quy trên có điểm dừng
 - (c) Hàm đệ quy trên sẽ xảy ra lỗi khi gọi hàm với tham số thực có giá trị âm
 - (d) Cả a, b, c đều sai
7. Danh sách liên kết không có đặc điểm nào dưới đây [2.3, 13]
 - (a) Khi thực hiện thao tác chèn hoặc xóa, không cần di chuyển các phần tử
 - (b) Không cần phải xin cấp phát bộ nhớ trước
 - (c) Tốc độ truy xuất các phần tử giống như nhau
 - (d) Lượng bộ nhớ cấp phát cho danh sách liên kết tỷ lệ thuận với chiều dài của danh sách
8. Sắp xếp dãy số sau {15, 9, 7, 8, 20, -1, 4}. Biết rằng sau một lần (one pass) sắp xếp, dãy số trên trở thành {4, 9, -1, 8, 20, 7, 15}, hãy cho biết phương pháp sắp xếp nào đã được sử dụng: [6.1, 18]
 - (a) Sắp xếp lựa chọn
 - (b) Sắp xếp nhanh (Quick sort)
 - (c) Shell sort
 - (d) Nổi bọt

9. Sử dụng giải thuật Prim để tìm cây phủ tối thiểu đối với đồ thị sau. Biết rằng bắt đầu từ đỉnh 0, hãy cho biết thứ tự các cạnh được chọn để thêm vào cây phủ tối thiểu. [7.7, 22]



- (a) (A, C) (D, F) (B, E) (C, F) (B, C)
 (b) (A, C) (C, F) (D, F) (B, C) (B, E)
 (c) (A, C) (D, F) (B, E) (B, C) (C, F)
 (d) (A, C) (B, C) (B, E) (C, F) (D, F)
10. Khi giải bài toán tháp Hanoi (Tower of Hanoi) bằng phương pháp đệ quy, nhận định nào dưới đây đúng khi chạy với giá trị tham số lớn: [1.4, 12]
- (a) Bài toán có độ phức tạp về không gian lớn
 (b) Bài toán có độ phức tạp về thời gian lớn
 (c) Bài toán có độ phức tạp về thời gian lẫn không gian đều lớn
 (d) Bài toán có độ phức tạp về thời gian lẫn không gian đều nhỏ
11. Nhận định nào dưới đây về heap KHÔNG ĐÚNG [4.1, 25]
- (a) Heap nhị phân là cây nhị phân đầy đủ hoặc gần đầy đủ
 (b) d-heap (heap mà mỗi node có d nhánh) thích hợp cho những ứng dụng mà số lượng thao tác chèn nhiều hơn so với xóa
 (c) Trong max-heap với n phần tử ($n \geq 3$), phần tử nhỏ nhất sẽ nằm ở nút lá
 (d) Trong 3 đáp án trên, có 2 đáp án là đúng

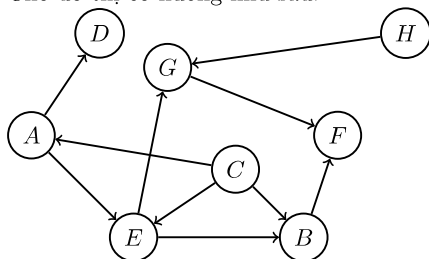
12. Giá trị của biểu thức tiền tố (prefix): [2.5, 24]
 $- + * 8 - 3 12 * + 2 9 7 3$

- (a) 1
 (b) 2
 (c) 3
 (d) Tất cả đều sai
13. Giả sử dùng mảng $A[0..m-1]$ để hiện thực hàng đợi vòng (circular queue), biết rằng vị trí của đầu và cuối là front và rear, thì [2.1, 14]
- (a) Có thể xảy ra trường hợp $\text{rear} < \text{front}$
 (b) Số lượng phần tử trong hàng đợi được tính theo công thức sau $(\text{rear} - \text{front} + m) \% m + 1$
 (c) Khi rear bằng $m - 1$ thì hàng đợi đầy
 (d) Trong 3 đáp án trên, có 2 đáp án đúng
14. Giả sử dùng dãy để biểu diễn heap, hãy cho biết trong những dãy số dưới đây, dãy nào là biểu diễn của heap. [4.2, 21]
- (a) 75,65,30,15,25,45,20,10
 (b) 75,65,45,10,30,25,20,15
 (c) 75,45,65,30,15,25,20,10
 (d) 75,45,65,10,25,30,20,15
15. Dùng phương pháp sắp xếp chèn trực tiếp để tiến hành sắp xếp các dãy sau theo thứ tự tăng dần, hãy cho biết dãy nào cần số lần so sánh ít nhất. [6.4, 16]
- (a) 94,32,40,90,80,46,21,69
 (b) 32,40,21,46,69,94,90,80
 (c) 21,32,46,40,80,69,90,94
 (d) 90,69,80,46,21,32,94,40

II. Điền kết quả (3.0)

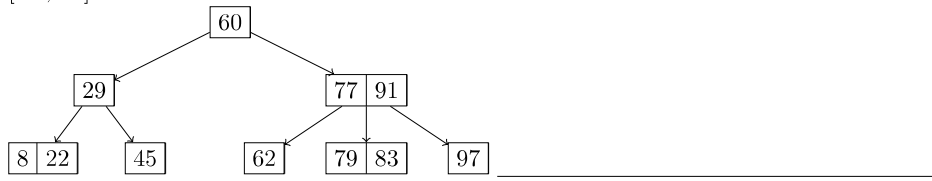
16. Cho cây nhị phân T, biết rằng mỗi node trong cây hoặc có 2 con, hoặc không có con nào. Chiều cao lớn nhất của T (có N nodes) có thể là _____. Tổng số node lớn nhất của T (chiều cao H) là _____ [3.1, 53]
17. Biểu thức $(a * b + c) / (d * e) + (f / g)$ biểu diễn dưới dạng hậu tố là [3.1, 52]:

18. Cho 2 hàm $f_1(n) = 100 + 2n$ và $f_2(n) = f_2(n/3 + 1)$, big-O của 2 hàm trên lần lượt là: [1.2, 51] _____
19. Cho đồ thị có hướng như sau:

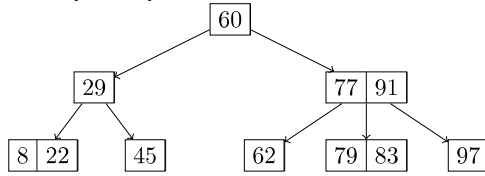


Cho biết thứ tự topo (topological order) dùng phép duyệt theo chiều sâu (thứ tự chọn đỉnh theo quy tắc tăng dần) [7.4, 55] _____

20. Cho cây B-tree bậc 3 như hình vẽ sau. Xóa khỏi cây 3 giá trị 62, 77, 79 (theo thứ tự). Vẽ trạng thái cây cuối cùng. [3.5, 59]

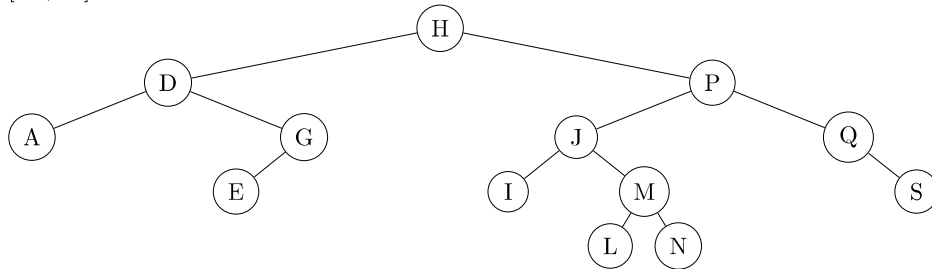


21. Cho cây B-tree bậc 3 như hình vẽ sau. Chèn vào cây 3 giá trị 72, 41, 39 (theo thứ tự trên). Vẽ trạng thái cây cuối cùng. [3.5, 58]



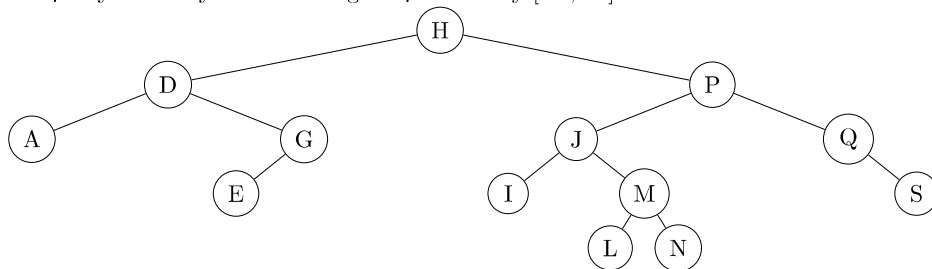
22. Cho dãy (22, 67, 13, 99, 4, 35, 44, 64, 17, 18). Chọn phần tử có index $N/2$ làm pivot. Sau bước chia đầu tiên: [6.1, 21]

23. Vẽ lại cây AVL này sau khi chèn giá trị **O** vào cây. Chỉ cần vẽ kết quả cuối cùng, không cần vẽ các bước trung gian [3.5, 56]



24. Cho heap sau: (65,43,59,24,37,48,57,12,23,28,5,3). Xóa phần tử lớn nhất ra khỏi heap, hãy cho biết heap kết quả (dưới dạng dãy): [4.4, 54]

25. Vẽ lại cây AVL này sau khi xóa giá trị **I** khỏi cây [3.5, 57]



III. Tự luận (5.0)

Chú ý: những phương thức cơ bản coi như có sẵn, không cần hiện thực lại. Ví dụ: heapUp, HeapDown, rotateLeft, ...

26. (2.0) Một hệ thống server đón nhận các record thông tin từ các thiết bị gửi về. Giả sử rằng chúng ta đã có cấu trúc **dRecord** chứa thông tin gửi về từ thiết bị.

- Server sử dụng một hàng đợi vòng để đón nhận các record gửi đến. Hãy hiện thực hàng đợi vòng chứa các record: **đặc tả hàng đợi**, hiện thực phương thức **int enqueue(dRecord* pData)** (trả về số slot còn trống trong queue). Lưu ý hàng đợi này có thể được khởi động với kích thước tùy biến. [2.4] (1.0)
- Một số thiết bị có độ ưu tiên khác nhau do đó các record được gửi đến sẽ có thêm một trường thông tin về độ ưu tiên, tính từ ID và timestamp. Hãy hiện thực một hàng đợi ưu tiên để lưu các record này: cấu trúc hàng đợi ưu tiên là gì? hiện thực các phương thức **int enqueue(dRecord* pData)**, **int dequeue(dRecord* pData)** cho hàng đợi sử dụng trường priority có trong record gửi về. [4.5] (1.0)

27. (1.0) Để sắp xếp dãy N phần tử ta có thể dùng nhiều giải thuật sắp xếp khác nhau.

- Hiện thực giải thuật quick sort với pivot là phần tử ngẫu nhiên trong dãy. [6.3] (0.5)
- Khi sử dụng quick sort trên dãy có N phần tử, giả sử rằng phân bố giá trị key là phân bố đều, tầm giá trị của key là $[1, M]$.
Hãy cho biết giải thuật quick sort chạy hiệu quả hay không khi: $M \ll N$, $M \approx N$, $M \gg N$ [6.4] (0.5)

28. (2.0) Với hệ thống xử lý các thông tin gửi về như mô tả ở phần trên (câu 26), các record đã xử lý cần được đưa vào một cấu trúc quản lý giúp việc tìm kiếm hiệu quả (câu này áp dụng tính điểm harmony với bài assignment 2).

- Có 2 thông tin cần được sắp xếp là ID và timestamp
- Kích thước các records lớn và không thể load lên bộ nhớ toàn bộ dữ liệu.
- Nhu cầu phân tích dữ liệu diễn ra thường xuyên và dữ liệu lỗi chỉ được xóa vào cuối ngày nếu cần.

Sử dụng cây AVL làm cấu trúc dữ liệu phục vụ tìm kiếm, các giá trị trên node của AVL là con trỏ đến cấu trúc **dRecord**. Có 2 cây AVL có thể sử dụng: một cây sắp xếp trên ID, một cây sắp xếp theo timestamp. Sinh viên hãy:

- Hãy hiện thực hàm tìm kiếm theo timestamp với tham số $(t_A, \Delta t_A)$. [3.6] (1.0)
bool find(AVLNode<dRecord*> *pR, int tA, int deltaT, vector<dRecord*> & ret);
- Hiện thực hàm tìm kiếm để lấy các record với ID và timestamp yêu cầu: $(ID, t_A, \Delta t)$ [3.7] (0.5)
bool find(AVLNode<dRecord*> *pR, int ID, int tA, int deltaT, vector<dRecord*> & ret);

Các phương thức (method) và hàm cần thiết đã học trong các cấu trúc dữ liệu cơ bản xem như đã có. Có thể sử dụng template nếu cần.

29. **[Bonus](+0.5)** Có một giải thuật sắp xếp sử dụng AVL làm cấu trúc trung gian. Phân tích tính hiệu quả của cách tiếp cận này (thời gian, không gian, độ phức tạp). [6.3]

Chủ nhiệm Khoa/Bộ môn

Giảng viên ra đề

Nguyễn Đức Dũng