



## KỲ THI CUỐI KỲ – HỌC KỲ 1 / 2019-2020

Môn thi: Cấu trúc dữ liệu & giải thuật [CO2003]  
GV ra đề: Vương Bá Thịnh, Nguyễn Đức Dũng  
Mã đề: 1915

Thời gian: 120'  
Ngày thi: 30-12-2019

Sinh viên được mang tài liệu giấy.

### I. Giải thuật cơ bản

1. Thực hiện các giải thuật sau (trình bày chi tiết từng bước):

- a [6.1] (1.0) Thực hiện Shellsort trên dãy số sau: 95, 90, 11, 30, 45, 21, 64, 96, 33, 57. Trong đó step là  $k = \{3, 1\}$ .
- b [4.4] (1.0) Dựng minHeap từ dãy số sau (sử dụng giải thuật  $O(N)$ ): 45, 95, 2, 89, 63, 0, 67, 28, 33, 78.
- c [6.4] (1.0) Trong các giải thuật sắp xếp đã thảo luận, giải thuật nào có đặc tính ổn định, giải thuật nào không? Giải thích.

2. Cho dãy số sau: 27, 33, 11, 85, xx, 51, 35, yy, 38, 62, 58. Trong đó 1abcdef là mã số sinh viên và  $xx = ef$ ,  $yy = ce$ .

- a [3.5] (0.5) Chèn các số trong dãy vào AVL theo đúng trình tự. Vẽ cây AVL qua các bước.
- b [3.5] (0.5) Xoá xx, yy ra khỏi cây theo trình tự.
- c [3.5] (1.0) Tạo B-tree bậc 3 từ dãy số trên.

### II. Tự luận

3. Cho một bàn cờ kích thước  $N \times N$ , trên đó chỉ có 2 loại quân cờ trắng và đen. Một nhóm định nghĩa như sau: 2 quân cờ cùng loại nằm cạnh nhau thuộc cùng 1 nhóm (nằm cạnh theo chiều ngang hoặc dọc, không tính đường chéo).

- a [2.4] (0.5) Hiện thực cấu trúc dữ liệu cần thiết cho bàn cờ này.
- b [8.4] (1.0) Hiện thực giải thuật tìm tất cả các quân cờ thuộc cùng nhóm bắt đầu từ 1 vị trí cho trước. Prototype của hàm như sau:  

```
void getGroup(Board2D &b, int posX, int posY, vector<pos> &rList);
```

  
Trong đó Board2D là kiểu dữ liệu lưu trữ bàn cờ, pos là kiểu cấu trúc chứa giá trị hàng, cột.

4. Một hệ thống hỗ trợ lái xe nhận các luồng dữ liệu camera từ các xe gửi về. Giả sử chúng ta đã có cấu trúc dữ liệu `iRecord` chứa dữ liệu gửi về, trong đó chứa thông tin về timestamp, vị trí chụp (longitude, latitude), vận tốc xe, và các thông số sensor khác đi kèm theo xe.

- a [3.6] (1.0) Để phục vụ truy vấn nhanh các record, người ta thực hiện đánh chỉ mục trên timestamp, vị trí (longitude, latitude), v.v. sử dụng cây AVL. Cho cây AVL sắp xếp trên timestamp, hãy hiện thực hàm tìm kiếm các record nằm trong khoảng sai số cho trước.  

```
bool find(AVLNode<iRecord*> *pR, int bTime, int eTime, vector<iRecord*> &ret);
```

  
trong đó bTime, eTime là thời gian bắt đầu và kết thúc của khoảng thời gian mà chúng ta cần truy vấn.
- b [5.5] (1.0) Để truy vấn vị trí nhanh hơn, chúng ta tiến hành đánh chỉ mục trên vị trí bằng cách chia lưới (longitude, latitude) thành các khoảng lưới ( $\Delta_{long}, \Delta_{lat}$ ). Với toạ độ bất kỳ ( $A_{long}, A_{lat}$ ), ta lượng hoá về giá trị chỉ mục ( $i_{long}, i_{lat}$ ). Từ toạ độ chỉ mục này, ta xây dựng hàm băm để truy vấn với các thông tin sau: kích thước bộ nhớ  $M$ , kích thước giới hạn không gian tìm kiếm ( $max_{long}, max_{lat}$ ). Sinh viên hãy phân tích hàm đề xuất, tỉ lệ đụng độ, cách giải quyết đụng độ. Hàm có dạng sau:  

```
void posHash(int i_long, int i_lat);
```

5. Để sắp xếp dãy  $N$  phần tử ta có thể dùng nhiều giải thuật sắp xếp khác nhau. Sinh viên hãy hiện thực giải thuật sắp xếp theo chiến lược sau:

- a [6.3] (0.5) Khi  $N > 1000$ , thực hiện MergeSort, ngược lại thì sử dụng QuickSort với pivot là phần tử ngẫu nhiên trong dãy.
- b [6.4] (0.5) Giả sử rằng phân bố giá trị key là giá trị nguyên phân bố đều, tầm giá trị của key là  $[1, M]$ . Giải thuật chạy hiệu quả khi nào, với giá trị nào của  $M$ .