

ĐỀ THI GIỮA KỲ THỰC HÀNH

Môn học: Cấu Trúc Dữ Liệu và Giải Thuật.

GV hướng dẫn thực hành: Nguyễn Khánh Toàn (ktoan271199@gmail.com).

1 Quy định nộp bài

- Bài làm cá nhân.
- Thời gian: 1 tuần (xem chi tiết trên Moodle).
- Thư mục bài làm: đặt tên là **<MSSV>**.
 - Báo cáo trình bày tóm tắt thuật toán giải quyết các bài toán (có thể chèn mã giả) và đánh giá độ phức tạp của thuật toán đề xuất một cách sơ lược (không cần chi tiết), định dạng pdf (đặt tên **<MSSV>.pdf**). Trang đầu tiên ghi thông tin sinh viên (tên thành viên, mã số sinh viên, lớp đang học).
 - Mã nguồn của mỗi chương trình đặt trong một thư mục riêng (Bai01, Bai02, Bai03); lưu ý chỉ nộp các file mã nguồn sử dụng để làm bài tập, xóa hết các file debug, release, v.v.
 - Ngôn ngữ: C hoặc C++.
- Nén thư mục bài làm thành định dạng zip rồi nộp trên Moodle.
- Báo cáo cần trình bày rõ ràng, mạch lạc; mã nguồn cần được tổ chức rõ ràng.
- Báo cáo/mã nguồn có tham khảo cần phải ghi nguồn rõ ràng ở cuối báo cáo.
- Các bài giống nhau (cho dù là một phần hay toàn bộ) sẽ bị 0 điểm. Các bài làm có mã nguồn chỉ khác nhau một phần ở tên biến, tên hàm cũng bị xem như là bài làm giống nhau.

2 Đề bài

2.1 Quy định về cài đặt

1. Trong các bài toán dưới đây, các bạn **không được sử dụng** các cấu trúc dữ liệu có sẵn của các thư viện như *list*, *stack*, *queue*, *priority queue*, *map*,

unordered_map, hay các cấu trúc dữ liệu nâng cao khác, chỉ được sử dụng hai thư viện là *vector* và *string* hoặc mảng động tự cài đặt.

2. Tương tự, các bạn sẽ **không được sử dụng** các hàm có sẵn của các thư viện để phục vụ mục đích sắp xếp và tìm kiếm như hàm *sort*, *lower_bound*, *upper_bound*, *binary_search*, v.v.
3. Các bạn sẽ phải tận dụng lại các bản cài đặt của mình ở các bài tập tuần trước để giải quyết.
4. Mã nguồn sẽ được chấm tự động, do đó các bạn bắt buộc phải tuân thủ nguyên tắc tổ chức đầu vào và đầu ra như yêu cầu của mỗi bài tập.

2.2 Bài 1 (3 điểm)

Cho một danh sách liên kết đơn chứa n số nguyên, các phần tử của danh sách đang được tổ chức như sau:

$$L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow \dots \rightarrow L_n.$$

Các bạn cài đặt chương trình tổ chức lại danh sách trên theo thứ tự mới dưới đây:

$$L_1 \rightarrow L_n \rightarrow L_2 \rightarrow L_{n-1} \rightarrow L_3 \rightarrow L_{n-2} \rightarrow \dots$$

Gợi ý: Các bạn nên bắt đầu bằng việc tìm nút chính giữa của danh sách, sau đó hãy xem xét phần danh sách từ nút chính giữa trở về sau.

2.2.1 Yêu cầu

- Phải sử dụng danh sách liên kết đơn tự cài đặt bằng con trỏ để tổ chức dữ liệu.
- Chỉ được dùng các phép hoán đổi liên kết trong danh sách liên kết đơn để thực hiện cài đặt thuật toán, không được thay đổi giá trị của các nút trong danh sách.
- Độ phức tạp về mặt không gian bộ nhớ phải đảm bảo là $O(1)$, nghĩa là các bạn không được tạo thêm một số lượng nút tuyến tính với N , chẳng hạn là không được tạo thêm $\frac{N}{2}$ nút mới với N là kích thước của danh sách.
- Nếu vi phạm một trong các yêu cầu trên các bạn sẽ bị trừ 50% số điểm mà các bạn đạt được của câu này.

2.2.2 Đầu vào

- Dòng đầu tiên quy định số phần tử của danh sách là N , thỏa mãn $1 \leq N \leq 10^3$.
- Dòng tiếp theo mô tả các phần tử trong danh sách ban đầu, mỗi phần tử a_i là một số nguyên thỏa mãn $1 \leq a_i \leq 10^6$.

2.2.3 Đầu ra

Đầu ra của chương trình là danh sách đã được sắp xếp lại theo thứ tự yêu cầu.

2.2.4 Ví dụ mẫu

Đầu vào:

```
4
3 6 8 10
```

Đầu ra:

```
3 10 6 8
```

2.3 Bài 2 (4 điểm)

Nhân dịp sinh nhật, Lazada tổ chức chương trình khuyến mãi trà sữa cho khách hàng. Mỗi lượt khuyến mãi sẽ có tất cả n ly trà sữa được chế biến, với giá tiền mỗi ly đã được khuyến mãi là \mathbf{b}_i . Do số lượng trà sữa có hạn, nên n vị khách ngẫu nhiên sẽ được lựa chọn để tham gia chương trình, và số tiền mỗi người có thể chi trả cho việc mua trà sữa là \mathbf{a}_i .

Ban đầu mỗi người chơi sẽ được chỉ định để mua ly trà sữa đúng với vị trí của mình. Tuy nhiên, để tăng tính tương tác cho khách hàng và đảm bảo công bằng, Lazada quyết định tổ chức khuyến mãi theo hình thức ghép cặp người chơi. Một cặp đôi 'tốt' được định nghĩa là tổng số tiền họ bỏ ra để mua trà sữa phải lớn hơn tổng giá tiền hai ly trà sữa tại vị trí tương ứng, hay nói cách khác là $\mathbf{a}_i + \mathbf{a}_j > \mathbf{b}_i + \mathbf{b}_j$ với hai người tham gia tại vị trí i và j .

Lazada muốn biết trước rằng có thể có bao nhiêu cặp đôi tốt tham gia được chương trình khuyến mãi. Lưu ý rằng mỗi người có thể ghép cặp được với nhiều người chơi khác và *không nhất thiết* phải có điều kiện: $a_i > b_i, \forall i : 1 \leq i \leq n$.

Các bạn cài đặt chương trình để giúp Lazada giải quyết vấn đề trên, sẽ có hai mức điểm cho câu hỏi này:

- (2 điểm) Cài đặt thuật toán có độ phức tạp là $O(n^2)$ - các bạn chỉ đúng được tối đa 50% bộ test.
- (4 điểm) Cài đặt thuật toán có độ phức tạp là $O(n \log n)$ - các bạn sẽ được tối đa số điểm của câu này nếu cài đặt đúng.

2.3.1 Đầu vào

- Dòng đầu tiên quy định số ly trà sữa được chế biến hay số người được tham gia chương trình, là số nguyên n thỏa $2 \leq n \leq 2 \cdot 10^5$.
- Dòng tiếp theo bao gồm n số nguyên $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^6)$, trong đó a_i là số tiền của mỗi người chơi bỏ ra để tham gia chương trình.

- Dòng tiếp theo bao gồm n số nguyên $b_1, b_2, \dots, b_n (1 \leq b_i \leq 10^6)$, trong đó b_i là giá tiền được khuyến mãi của mỗi ly trà sữa.

2.3.2 Đầu ra

Yêu cầu đầu ra của chương trình là tổng số cặp đôi 'tốt' có thể tham gia khuyến mãi, là một số nguyên.

2.3.3 Ví dụ mẫu

Ví dụ 1. Đầu vào:

```
4
3 2 4 5
2 3 6 4
```

Đầu ra:

```
1
```

Giải thích: Trong ví dụ trên, chỉ có một cặp đôi tốt là người chơi thứ 1 và người chơi cuối cùng, thỏa mãn $a_1 + a_4 = 8 > 6 = b_1 + b_4$.

Ví dụ 2. Đầu vào:

```
4
3 2 4 5
2 2 6 4
```

Đầu ra:

```
3
```

Giải thích: Trong ví dụ trên, có ba cặp đôi tốt là người chơi thứ 1 và người chơi cuối cùng, thỏa mãn $a_1 + a_4 = 8 > 6 = b_1 + b_4$; người chơi thứ 1 và người chơi thứ hai, thỏa mãn $a_1 + a_2 = 5 > 4 = b_1 + b_2$; người chơi thứ 2 và người chơi cuối cùng, thỏa mãn $a_2 + a_4 = 7 > 6 = b_2 + b_4$.

2.4 Bài 3 (3 điểm)

Cho trước một mảng số nguyên A ngẫu nhiên có kích thước N , các bạn hãy thực hiện xây dựng một cây tìm kiếm nhị phân có chiều cao cân bằng từ các phần tử trong mảng. *Cây tìm kiếm nhị phân có chiều cao cân bằng* được định nghĩa là với mỗi nút, sự khác biệt giữa chiều cao của cây con ở hai bên trái và

phải của nó tối đa là 1. Lưu ý các bạn chỉ xem xét mảng ban đầu **không có phần tử nào trùng nhau** (testcase mình thiết kế sẽ ràng buộc điều kiện này).

2.4.1 Yêu cầu

- Không thực hiện các thao tác xoay cây trong quá trình xây dựng.
- Thuật toán đề xuất có độ phức tạp tối đa là $O(n \log n)$.
- Nếu vi phạm một trong các yêu cầu trên các bạn sẽ bị trừ 50% số điểm mà các bạn đạt được của câu này.
- Lưu ý rằng các bạn có thể thực hiện sắp xếp lại thứ tự các phần tử trong mảng ban đầu để thực hiện yêu cầu trên và có thể có nhiều thuật toán xây dựng cây thỏa mãn yêu cầu bài toán.

2.4.2 Đầu vào

- Dòng đầu tiên quy định số phần tử của mảng N , thỏa mãn $1 \leq N \leq 10^5$.
- Dòng tiếp theo bao gồm n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$).

2.4.3 Đầu ra

Đầu ra của chương trình là thứ tự **pre-order** (thứ tự duyệt trước) và thứ tự **in-order** (thứ tự duyệt giữa) của cây nhị phân tìm kiếm mà bạn xây dựng; được in ra theo thứ tự và thể hiện bằng 2 dòng cách nhau bởi dấu xuống dòng. Chẳng hạn:

```
2 1 3
1 2 3
```

Giải thích: Dòng đầu tiên và dòng thứ hai của ví dụ trên là thứ tự duyệt trước và duyệt giữa của cây nhị phân tìm kiếm được xây dựng.

Hết