

# DANH SÁCH KIỂM TRA THỰC HÀNH

**Môn:** Cấu trúc dữ liệu và giải thuật

**Thời gian:** 30 phút

## A. PHẦN I: (4 điểm)

**Câu 1:** Xét các thuật toán có thời gian chạy như bên dưới. Hỏi mỗi thuật toán chậm đi bao nhiêu lần khi gấp đôi kích thước đầu vào?

$n$

$n^2$

$n^3$

$100n^2$

$n \log n$

$2^n$

**Câu 2:** Sắp xếp các hàm sau đây theo thứ tự tăng dần của tốc độ tăng; nghĩa là, hàm  $f(n)$  sẽ được xếp trước hàm  $g(n)$  nếu  $f(n) = O(g(n))$ .

$f_1(n) = n^{2,5}$

$f_2(n) = \sqrt{2n}$

$f_3(n) = n + 10$

$f_4(n) = 10^n$

$f_5(n) = 100^n$

$f_6(n) = n^2 \log n$

**Câu 3:** Phân tích thời gian chạy (dùng ký hiệu  $O$ ) của thuật toán tìm phần tử lớn nhất.

```
max ← a0
```

```
for i ← 1 to n-1
```

```
    if (ai > max)
```

```
        max ← ai
```

**Câu 4:** Phân tích thời gian chạy (dùng ký hiệu  $O$ ) của các đoạn chương trình C++ sau đây.

(a) `sum = 0;`

`for (i = 0; i < n; i++)`

`sum++;`

(b) `sum = 0;`

`for (i = 0; i < n; i++)`

`for (j = 0; j < n; j++)`

`sum++;`

**Câu 5:** Phân tích thời gian chạy (dùng ký hiệu  $O$ ) của các đoạn chương trình C++ sau đây.

(a) `sum = 0;`

`for (i = 0; i < n; i++)`

`for (j = 0; j < n*n; j++)`

`sum++;`

(b) `sum = 0;`

```

for (i = 0; i < n; i++)
    for (j = 0; j < i; j++)
        sum++;

```

**Câu 6:** Phân tích thời gian chạy (dùng ký hiệu  $O$ ) của các đoạn chương trình C++ sau đây.

- (a) `sum = 0;`  
`for (i = 0; i < n; i++)`  
 `for (j = 0; j < i*i; j++)`  
 `for (k = 0; k < j; k++)`  
 `sum++;`
- (b) `sum = 0;`  
`for (i = 0; i < n; i++)`  
 `for (j = 0; j < i*i; j++)`  
 `if (j % i == 0)`  
 `for (k = 0; k < j; k++)`  
 `sum++;`

**Câu 7:** Viết chương trình thực hiện công việc sau:

- Xét một vector đang có kích thước  $s_1$  và dung lượng  $c_1$ , trong đó  $s_1 \leq c_1$ . Nêu các bước phải thực hiện để tăng dung lượng vector từ  $c_1$  lên  $c_2$ , trong đó  $c_1 < c_2$ . Sau khi tăng dung lượng như vậy thì kích thước vector là bao nhiêu?
- Xét một vector đang chứa các phần tử như sau:

{ 6, 5, 8, 2, 9, 7 }

Giả thiết rằng vector chưa đầy và vị trí của các phần tử tính từ 0. Nêu các bước phải thực hiện để chèn giá trị  $X$  vào vị trí 3 trong vector.

**Câu 8:** Viết chương trình thực hiện nhiệm vụ sau:

- Xét một vector đang chứa các phần tử như sau:  
 { 8, 1, 9, 3, 4, 6 }  
 Nêu các bước phải thực hiện để xóa phần tử ở vị trí 2 trong vector (vị trí tính từ 0).
- Hỏi giữa chèn/xóa ở đầu vector và chèn/xóa ở cuối vector thì thao tác nào chạy nhanh hơn? Vì sao?
- Giả sử ta phải bổ sung thao tác truncate vào vector nhằm cắt bỏ phần dung lượng dư thừa. Hãy đề xuất các bước cụ thể để thực hiện thao tác truncate.

**Câu 9:** Vì sao hàm hủy của danh sách liên kết đơn tốn thời gian chạy  $O(n)$ , trong khi của vector chỉ là  $O(1)$ ? Hãy đề xuất các thuật toán chèn/xóa ở cuối danh sách liên kết đơn? So sánh thời gian chạy với các thao tác chèn/xóa ở đầu danh sách? Trong trường hợp thuật toán đề xuất chạy chậm hơn, có cách nào làm cho nó chạy nhanh hơn không?

**Câu 10:** Hãy đề xuất một thuật toán truy nhập phần tử của danh sách liên kết đơn thông qua chỉ số. So sánh thời gian chạy của thuật toán này với thuật toán tương ứng của vector.

**Câu 11:** Vì sao chèn/xóa ở giữa danh sách liên kết (đơn, đôi hoặc vòng tròn) chạy nhanh hơn chèn/xóa ở giữa vector?

**Câu 12:** Nêu một ứng dụng của danh sách liên kết vòng tròn. Phải điều chỉnh ứng dụng đó như thế nào nếu dùng danh sách liên kết đơn thay cho danh sách liên kết vòng tròn?

**Câu 13:** Nêu cách thực hiện các thao tác lấy kích thước và kiểm tra rỗng của ngăn xếp trong các trường hợp sau:

(a) Cài đặt ngăn xếp bằng danh sách liên kết đơn.

(b) Cài đặt ngăn xếp bằng mảng.

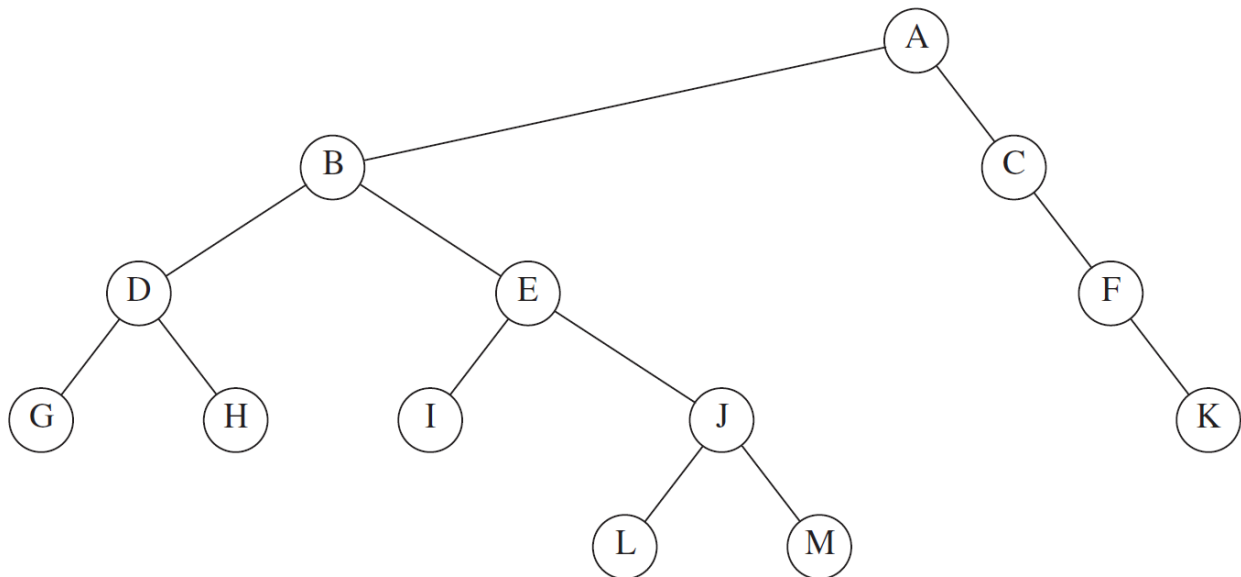
**Câu 14:** Đề xuất cách thực hiện hai ngăn xếp dùng chung một mảng.

**Câu 15:** Đề xuất cấu trúc dữ liệu hỗ trợ hai thao tác push và pop, như trong trường hợp ngăn xếp, nhưng hỗ trợ thêm thao tác findMin trả về phần tử nhỏ nhất trong thời gian  $O(1)$ .

**Câu 16:** Xét một hàng đợi kiểu mảng vòng tròn. Hãy chỉ ra dãy thao tác enqueue và dequeue phải thực hiện để chuyển hàng đợi từ trạng thái 1 sang trạng thái 2:



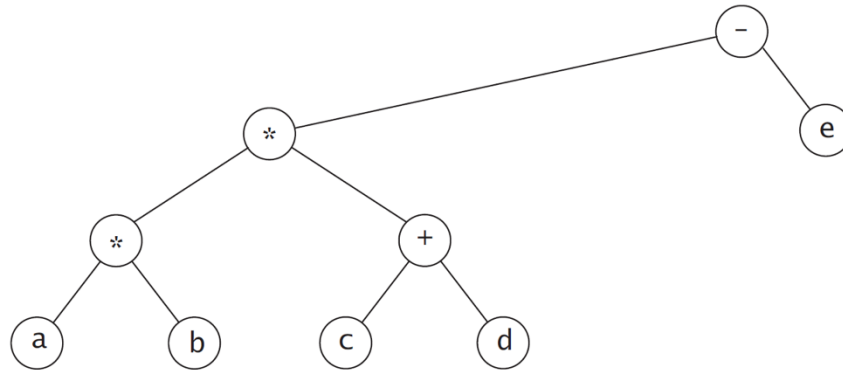
**Câu 17:** Chỉ ra nút gốc và các nút lá trên cây. Tính chiều sâu và chiều cao của các nút C, E và H. Tính chiều sâu và chiều cao của cây.



**Câu 18:** Xét cây nhị phân có  $n$  nút. Chứng minh rằng có  $n + 1$  liên kết rỗng (tức là con trỏ NULL) trên cây.

**Câu 19:** Xét cây nhị phân có chiều cao  $h$ . Chứng minh rằng số nút trên cây không vượt quá  $2^{h+1} - 1$ . Gọi ý: Dùng quy nạp toán học.

**Câu 20:** Viết các biểu thức trung tố, tiền tố và hậu tố tương ứng với cây biểu thức bên dưới.



**Câu 21:** Viết hàm C++ có một tham số duy nhất là con trỏ tới gốc của một cây nhị phân T để:

- (a) đếm số nút của T.
- (b) đếm số nút lá của T.
- (c) đếm số nút có đủ cả 2 con của T.

Gợi ý: Dùng đệ quy.

**Câu 22:** Viết các hàm C++ duyệt cây nhị phân theo thứ tự trước, giữa và sau. Gợi ý: Dùng đệ quy.

**Câu 23:** Nêu cách thực hiện phép duyệt cây nhị phân theo **thứ tự mức**: thăm nút gốc trước tiên, rồi đến thăm các nút ở độ sâu 1, rồi đến thăm các nút ở độ sâu 2, v.v... Gợi ý: Dùng hàng đợi.

**Câu 24:** Chèn lần lượt các giá trị sau đây vào cây nhị phân tìm kiếm đang rỗng: 20, 15, 19, 26, 31, 21, 14, 23, 25. Sau đó, xóa nút gốc của cây.

**Câu 25:** Viết hàm nhận vào một cây nhị phân tìm kiếm và hai giá trị  $k_1$  và  $k_2$ , trong đó  $k_1 \leq k_2$ . Hàm sẽ in ra tất cả các giá trị trên cây nằm trong khoảng  $[k_1; k_2]$ .

**Câu 26:** Vẽ hình mô tả quá trình biến đổi của một cây AVL đang rỗng khi chèn lần lượt vào cây các giá trị sau vào cây:

{ 2, 1, 4, 5, 9, 3, 6, 7 }

**Câu 27:** Xét bảng băm đang rỗng và có hàm băm là  $\text{hash}(x) = x \% 10$ . Hãy chèn lần lượt vào bảng các giá trị { 4371, 1323, 6173, 4199, 4344, 9679, 1989 } cho mỗi trường hợp sau:

- (a) Bảng băm dây chuyền
- (b) Bảng băm thăm dò tuyến tính
- (c) Bảng băm thăm dò bậc hai

**Câu 28:** Hãy tổ chức lại các bảng băm sau khi chèn trong bài tập 1 sao cho kích thước bảng băm gấp đôi.

**Câu 29:** Xét một đồng đang rỗng. Hãy chèn (insert) lần lượt vào đồng các giá trị sau đây: { 10, 12, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, 2 }

- b. Xây dựng đồng dùng hàm khởi tạo từ danh sách giá trị cho trong câu a.

**Câu 30:** Thực hiện thao tác xóa (deleteMin) 3 lần liên tiếp trên các đồng thu được trong câu 29.

## B. PHẦN 2 (6 điểm)

**Câu 1:** Viết chương trình sử dụng vector trong C++ thực hiện các công việc sau:

1. Nhập dữ liệu cho vector bằng cách chèn vào 1 vị trí pos bất kỳ
2. Xoá phần tử ở vị trí pos bất kỳ
3. Hiển thị vector theo chiều thuận
4. Hiển thị vector theo chiều ngược

**Câu 2:** Viết chương trình sử dụng vector trong C++ thực hiện các công việc sau:

1. Nhập dữ liệu cho vector
2. Nhập dữ liệu cho vector bằng cách chèn vào 1 vị trí pos bất kỳ
3. Xoá phần tử ở vị trí pos bất kỳ
4. Hiển thị giá trị lớn nhất và nhỏ nhất của vector

**Câu 3:** Viết chương trình sử dụng vector trong C++ thực hiện đọc vào n số. Đọc thêm một số x, chèn số x vào vị trí thứ k của dãy với k đọc vào từ bàn phím. Nếu  $k > n$  thì thêm vào vị trí thứ n. Đưa vector sau khi chèn ra màn hình.

**Câu 4:** Tạo một struct Nhân viên có 2 trường Họ tên, Lương và tuổi. Thực hiện công việc sau:

1. Nhập thông tin vào danh sách n nhân viên (liên kết đơn), n được nhập vào từ bàn phím.
2. Hiển thông tin của các nhân viên có mức lương cao nhất .
3. Hiển thị kết quả sau khi nhập

**Câu 5:** Tạo một struct Nhân viên có 2 trường Họ tên, Lương và tuổi. Thực hiện công việc sau:

1. Nhập thông tin vào danh sách n nhân viên (liên kết đôi), n được nhập vào từ bàn phím.
2. Hiển thông tin của các nhân viên có mức lương cao nhất .
3. Hiển thị kết quả sau khi nhập

**Câu 6:** Tạo một struct Nhân viên có 2 trường Họ tên, Lương và tuổi. Thực hiện công việc sau:

1. Nhập thông tin vào danh sách n nhân viên (liên kết vòng), n được nhập vào từ bàn phím.
2. Hiển thông tin của các nhân viên có mức lương cao nhất .
3. Hiển thị kết quả sau khi nhập

**Câu 7:** Tạo một struct Nhân viên có 2 trường Họ tên, Lương và tuổi. Thực hiện công việc sau:

1. Nhập thông tin vào danh sách n nhân viên (liên kết đơn), n được nhập vào từ bàn phím.
2. Chèn thông tin của các nhân viên mới vào giữa theo vị trí pos .
3. Hiển thị kết quả sau khi chèn

**Câu 8:** Tạo một struct Nhân viên có 2 trường Họ tên, Lương và tuổi. Thực hiện công việc sau:

1. Nhập thông tin vào danh sách n nhân viên (liên kết đôi), n được nhập vào từ bàn phím.
2. Chèn thông tin của các nhân viên mới vào giữa theo vị trí pos .
3. Hiển thị kết quả sau khi chèn

**Câu 9:** Viết chương trình thực hiện công việc sau:

1. Chèn vào hàng đợi một số phần tử
2. Hiển thị các phần tử trong hàng đợi
3. Kiểm tra giá trị x có trong hàng đợi không, x được nhập vào từ bàn phím

**Câu 10:** Viết chương trình thực hiện công việc sau:

1. Chèn vào hàng đợi một số phần tử
2. Rút từng phần tử khỏi hàng đợi
3. Hiển thị các phần tử trong hàng đợi

**Câu 11:** Viết chương trình thực hiện công việc sau:

1. Chèn vào ngăn xếp một số phần tử
2. Hiển thị các phần tử trong ngăn xếp
3. Kiểm tra giá trị x có trong ngăn xếp không, x được nhập vào từ bàn phím

**Câu 12:** Viết chương trình thực hiện công việc sau:

1. Chèn vào ngăn xếp một số phần tử
2. Rút từng phần tử khỏi ngăn xếp
3. Hiển thị các phần tử trong ngăn xếp

**Câu 13:** Viết chương trình thực hiện công việc sau:

1. Tạo 1 cây tìm kiếm nhị phân
2. Kiểm tra có phần tử có giá trị x trong cây không, x nhập vào từ bàn phím
3. Duyệt cây theo thứ tự trước, giữa, sau

**Câu 14:** Viết chương trình thực hiện công việc sau:

1. Tạo 1 cây tìm kiếm nhị phân
2. Chèn 1 phần tử vào cây, x nhập vào từ bàn phím
3. Duyệt cây theo thứ tự trước, giữa, sau

**Câu 15:** Tạo một struct Nhân viên có 2 trường Họ tên, Lương và tuổi. Thực hiện công việc sau:

4. Nhập thông tin vào danh sách n nhân viên (liên kết vòng), n được nhập vào từ bàn phím.
5. Xoá nhân viên ở vị trí pos, pos nhập vào từ bàn phím
6. Hiển thị kết quả sau khi xoá.

**Câu 16:** Tạo một struct Nhân viên có 2 trường Họ tên, Lương và tuổi. Thực hiện công việc sau:

1. Nhập thông tin vào danh sách n nhân viên (liên kết đơn), n được nhập vào từ bàn phím.
2. Xoá nhân viên ở vị trí pos, pos nhập vào từ bàn phím
3. Hiển thị kết quả sau khi xoá.

**Câu 17:** Tạo một struct Nhân viên có 2 trường Họ tên, Lương và tuổi. Thực hiện công việc sau:

1. Nhập thông tin vào danh sách n nhân viên (liên kết đôi), n được nhập vào từ bàn phím.
2. Xoá nhân viên ở vị trí pos, pos nhập vào từ bàn phím
3. Hiển thị kết quả sau khi xoá.

**Câu 18:** Giả sử có chuỗi ký tự myString = “Chao 64TTNT\*\*\*\* của Đại học Thủy lợi !\*\*\*\* ”, hãy viết chương trình thực hiện công việc sau:

1. Kiểm tra chuỗi myString nếu gặp \* thì thực pop() của ngăn xếp.
2. Kiểm tra chuỗi myString nếu không gặp ký tự \* thì thực push() của ngăn xếp.
3. Hiển thị ngăn xếp sau khi thực hiện thao tác 1,2.

**Câu 19:** Sắp xếp dãy { 3, 1, 4, 1, 5, 9, 2, 6, 5 } dùng sắp xếp chọn.

**Câu 20:** Sắp xếp dãy { 3, 1, 4, 1, 5, 9, 2, 6, 5 } dùng sắp xếp nổi bọt.

**Câu 21:** Sắp xếp dãy { 3, 1, 4, 1, 5, 9, 2, 6, 5 } dùng sắp xếp chèn.

**Câu 22:** Dùng thuật toán sắp xếp vun đống để sắp xếp dãy:

{ 142, 543, 123, 65, 453, 879, 572, 434, 111, 242, 811, 102 }

**Câu 23:** Dùng thuật toán sắp xếp trộn để sắp xếp dãy:

{ 3, 1, 4, 1, 5, 9, 2, 6 }

**Câu 24:** Dùng thuật toán sắp xếp nhanh để sắp xếp dãy:

{ 7, 6, 1, 5, 2, 8, 4, 3 }

**Câu 25:** Các bài thực hành về hàm băm, bảng băm

**Câu 26:** Các bài thực hành về sắp xếp

**Câu 27:** Các bài thực hành về hàng đợi ưu tiên

**Câu 28:** Các bài thực hành về Cây sắp xếp nhị phân.