

[Trang chủ](#) / [Các khoá học của tôi](#) / [Học kỳ 2 \(2023-2024\)](#) / [Khoa Khoa Học Máy Tính](#) / [IT003.O21](#) / [General](#) / [Quiz 3: Quick sort](#)

Bắt đầu vào lúc	Thứ Năm, 14 tháng 3 2024, 10:42 PM
Trạng thái	Đã xong
Kết thúc lúc	Thứ Năm, 14 tháng 3 2024, 10:44 PM
Thời gian thực hiện	1 phút 32 giây
Điểm	8,00 trên 10,00 (80%)

Câu hỏi 1

Đúng

Đạt điểm 1,00 trên 1,00

Thuật toán nào sau đây THUỘC lớp stable sorting algorithms?

(Stable sorting algorithms maintain the relative order of records with equal keys (i.e. values). That is, a sorting algorithm is stable if whenever there are two records R and S with the same key and with R appearing before S in the original list, R will appear before S in the sorted list.)

- ☐ a. Quick sort
- ☐ b. Heap sort
- ☒ c. Merge sort ✓
- ☐ d. Interchange sort

Câu trả lời của bạn đúng

The correct answer is: Merge sort

Câu hỏi 2

Đúng

Đạt điểm 1,00 trên 1,00

Thuật toán nào bên dưới sử dụng chiến lược chia để trị (divide and conquer)?

GHI CHÚ: CÂU HỎI NÀY CÓ THỂ CHỌN NHIỀU ĐÁP ÁN (NẾU CÓ).

- ☐ a. Heap sort
- ☐ b. Insertion sort
- ☒ c. Quick sort ✓
- ☒ d. Merge sort ✓

Câu trả lời của bạn đúng

The correct answers are: Merge sort, Quick sort

Câu hỏi 3

Đúng

Đạt điểm 1,00 trên 1,00

Hãy viết hàm **partition** để hoàn chỉnh thuật toán Quick sort sắp xếp tăng dần cho mảng 1 chiều n phần tử.

```
void quick_sort_ascending(int *a, int left, int right){
    if (left>=right) return;
    int i, j;
    partition(a, left, right, i, j);
    quick_sort_ascending(a, left, j);
    quick_sort_ascending(a, i, right);
}

void quick_sort_ascending(int *A, int N){
    quick_sort_ascending(A, 0, N-1);
}
```

<p>A.</p> <pre>void partition(int *a, int l, int r, int i, int j){ int pivot = a[(l+r)/2]; i=l; j=r; while(i<=j) { if(a[i] > pivot) i++; if(a[j] < pivot) j--; if(i<=j) { hoanvi(a[i], a[j]); i++; j--; } } }</pre>	<p>B.</p> <pre>void partition(int *a, int l, int r, int &i, int &j){ int pivot = a[(l+r)/2]; i=l; j=r; while(i<=j) { while(a[i] < pivot) i++; while(a[j] > pivot) j--; if(i<=j) { hoanvi(a[i], a[j]); i++; j--; } } }</pre>
<p>C.</p> <pre>void partition(int *a, int l, int r, int i, int j){ int pivot = a[(l+r)/2]; i=l; j=r; while(i<=j) { while(a[i]< pivot) i++; while(a[j]> pivot) j--; if(i<=j) { hoanvi(a[i], a[j]); i++; j--; } } }</pre>	<p>D.</p> <pre>void partition(int *a, int l, int r, int &i, int &j){ int pivot = a[(l+r)/2]; i=l; j=r; while(i<=j) { while(a[i] > pivot) i++; while(a[j] < pivot) j--; if(i<=j) { hoanvi(a[i], a[j]); i++; j--; } } }</pre>

- ☐ a. Hình C
- ☐ b. Hình A
- ☒ c. Hình B ✓

- ☐ d. Hình D

Câu trả lời của bạn đúng

The correct answer is: Hình B

Câu hỏi 4

Đúng

Đạt điểm 1,00 trên 1,00

Giả sử dùng thuật toán Quick sort để sắp xếp 1000 phần tử, tốn ít nhất 200ms. Hỏi rằng cần ít nhất bao nhiêu thời gian để sắp xếp 200 phần tử?

- ☐ a. 60.2 ms
- ☐ b. 45.54 ms
- ☒ c. 31.11 ms ✓
- ☐ d. 20 ms

Câu trả lời của bạn đúng

The correct answer is: 31.11 ms

Câu hỏi 5

Đúng

Đạt điểm 1,00 trên 1,00

Độ phức tạp thời gian trong trường hợp tốt nhất của 4 thuật toán bên dưới được so sánh như sau:

(chú ý dấu so sánh, ví dụ, $a > b$: là độ phức tạp của a lớn hơn b, nghĩa là b hiệu quả hơn a)

- ☐ a. Merge sort > Quick sort > Insertion sort > Selection sort
- ☒ b. Insertion sort < Quick sort < Merge sort < Selection sort ✓
- ☐ c. Merge sort > Quick sort > Selection sort > Insertion sort
- ☐ d. Merge sort > Selection sort > Quick sort > Insertion sort

Câu trả lời của bạn đúng

The correct answer is: Insertion sort < Quick sort < Merge sort < Selection sort

Câu hỏi 6

Đúng

Đạt điểm 1,00 trên 1,00

Giả sử cần sắp xếp q dãy gồm 8 giá trị bằng thuật toán Quick sort. Thuật toán vừa hoàn thành lần phân hoạch đầu tiên với dãy số kết quả như sau: 1 6 3 9 10 15 20 16

- ☒ a. Phần tử pivot có thể là 9 hoặc 10 ✓
- ☐ b. Phần tử pivot có thể là 9
- ☐ c. Không phải 9, cũng không phải 10
- ☐ d. Phần tử pivot có thể là 10

Câu trả lời của bạn đúng

The correct answer is: Phần tử pivot có thể là 9 hoặc 10

Câu hỏi 7

Đúng

Đạt điểm 1,00 trên 1,00

Giả sử dùng thuật toán Quick sort để sắp xếp 1000 phần tử, tốn ít nhất 100ms. Hỏi rằng cần ít nhất bao nhiêu thời gian để sắp xếp 100 phần tử?

- ☒ a. 6.7 ms ✓
- ☐ b. 11.2 ms
- ☐ c. 50.2 ms
- ☐ d. 72.7 ms

Câu trả lời của bạn đúng

The correct answer is: 6.7 ms

Câu hỏi 8

Đúng

Đạt điểm 1,00 trên 1,00

Sắp xếp dãy số [30, 25, 40, 10, 15, 50, 20] tăng dần bằng thuật toán Quick sort. Chọn pivot là giá trị chính giữa mảng. Hãy cho biết sau khi thực hiện partition **đầu tiên** thì thứ tự các giá trị trong danh sách sẽ như thế nào?

```
void quick_sort(int *a, int left, int right){  
    int pivot=a[(left+right)/2];  
  
    int i=left, j=right;  
  
    while(i<=j){  
        while(a[i]< pivot) i++;  
        while(a[j]> pivot) j--;  
        if(i<=j) {  
            hoanvi(a[i], a[j]);  
            i++;j--;  
        }  
    }  
  
    if(left<j) quick_sort_ascending(a, left, j);  
    if(i<right) quick_sort_ascending(a, i, right);  
}
```

- ☒ a. 10, 25, 40, 30, 15, 50, 20 ✓
- ☐ b. 20, 10, 40, 25, 15, 50, 30
- ☐ c. 10, 25, 30, 40, 15, 50, 20
- ☐ d. 10, 25, 40, 15, 50, 20, 30

Câu trả lời của bạn đúng

The correct answer is: 10, 25, 40, 30, 15, 50, 20

Câu hỏi 9

Sai

Đạt điểm 0,00 trên 1,00

Sắp xếp dãy số $S=[30, 25, 40, 10, 15, 50, 20]$ tăng dần bằng thuật toán Quick sort (chọn pivot là giá trị chính giữa mảng, thuật toán Quick sort như trong tài liệu slide môn học). Thực hiện partion trên S ta được 2 mảng con $S1$ và $S2$. Tiếp tục thực hiện partion trên đoạn con $S1$ ta được kết quả đoạn $S1$ là:

```
void quick_sort(int *a, int left, int right){
    int pivot=a[(left+right)/2];

    int i=left, j=right;

    while(i<=j){
        while(a[i]< pivot) i++;
        while(a[j]> pivot) j--;
        if(i<=j) {
            hoanvi(a[i], a[j]);
            i++;j--;
        }
    }

    if(left<j) quick_sort_ascending(a, left, j);
    if(i<right) quick_sort_ascending(a, i, right);
}
```

- ☐ a. $S1=[10]$
- ☐ b. $S1$ không được thực hiện phép partition do chỉ tồn tại 1 phần tử
- ☐ c. $S1=[25, 10]$
- ☒ d. $S1=[10, 25]$ ✖

Câu trả lời của bạn sai.

The correct answer is: $S1$ không được thực hiện phép partition do chỉ tồn tại 1 phần tử

Câu hỏi 10

Sai

Đạt điểm 0,00 trên 1,00

Thuật toán Quick sort:

```
void quick_sort(int *a, int left, int right){
    int pivot=a[(left+right)/2];

    int i=left, j=right;

    while(i<=j){
        while(a[i]< pivot) i++;
        while(a[j]> pivot) j--;
        if(i<=j) {
            hoanvi(a[i], a[j]);
            i++;j--;
        }
    }

    if(left<j) quick_sort_ascending(a, left, j);
    if(i<right) quick_sort_ascending(a, i, right);
}
```

Hãy sắp xếp dãy số $S=[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]$ tăng dần bằng thuật toán Quick sort. Thực hiện partition trên S ta được 2 mảng con $S1$ và $S2$. Tiếp tục thực hiện partition trên đoạn con $S2$ ta được kết quả đoạn $S2$ là:

- ☐ a. $S2=[6, 7, 5, 9, 8]$
- ☐ b. $S2=[6, 7, 5, 8, 9]$
- ☒ c. $S2=[6, 7, 8, 9, 5]$ ✖
- ☐ d. $S2$ không được thực hiện phép partition do chỉ tồn tại 1 phần tử

Câu trả lời của bạn sai.

The correct answer is: $S2=[6, 7, 5, 9, 8]$ [◀ Quiz 2: Selection Insertion sort \(Deadline 10/3\)](#)

Chuyển tới...

[Quiz 4: Merge sort ▶](#)