



# QUICK SORT – QUEUE

1. Hồ Thái Ngọc
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang



# BÀI TOÁN

# Bài toán



- Hãy cài đặt thuật toán Quick Sort sắp xếp mảng một chiều các phân số bằng hai phương pháp.
  - + **Đệ quy**
  - + Không đệ quy bằng cách sử dụng kỹ thuật Queue.



# QUICK SORT ĐỆ QUY

# Hàm cài đặt thuật toán quicksort



— Hàm cài đặt

```
11. void QuickSort(PHANSO a[], int n)
12. {
13. |     QuickSort(a, 0, n-1);
14. }
```

# Hàm cài đặt thuật toán quicksort



— Hàm cài đặt

```
11. void QuickSort(PHANSO a[], int Left, int Right)
12. {
13.     if (Left < Right)
14.     {
15.         int iPivot = Partition(a, Left, Right);
16.         QuickSort(a, Left, iPivot - 1);
17.         QuickSort(a, iPivot + 1, Right);
18.     }
19. }
```

```
11.int Partition(PHANSO a[], int Left, int Right)
12.{
13.    PHANSO pivot = a[Right];
14.    int vt = (Left - 1);
15.    for (int i=Left; i<=Right-1; i++)
16.        if (SoSanh(a[i], pivot)==-1)
17.        {
18.            vt++;
19.            HoanVi(a[vt], a[i]);
20.        }
21.    vt = vt + 1;
22.    HoanVi(a[vt], a[Right]);
23.    return vt;
24.}
```





# QUICK SORT KHỬ ĐỆ QUY



# Quick sort khử đệ quy



```
1. struct phanso
2. {
3.     int tu;
4.     int mau;
5. };
6. typedef struct phanso PHANSO;
```

# Quick sort khử đệ quy



```
7. struct segment
8. {
9.     int Left;
10.    int Right;
11.};
12. typedef struct segment SEGMENT;
```

# Quick sort khủ đệ quy



```
13.struct queue
14.{
15.    int n;
16.    SEGMENT a[10000];
17.};
18.typedef struct queue QUEUE;
```

# Quick sort khử đệ quy



```
19.int Nhap(PHANSO[], int&, string);
```

```
20.int Xuat(PHANSO[], int, string);
```

```
21.void Init(QUEUE&);
```

```
22.int IsEmpty(QUEUE);
```

```
23.int IsFull(QUEUE);
```

```
24.void EnQueue(QUEUE&, SEGMENT);
```

```
25.SEGMENT DeQueue(QUEUE&);
```

# Quick sort khử đệ quy



```
26.void HoanVi(PHANSO&, PHANSO&);  
27.int SoSanh(PHANSO, PHANSO);  
28.int Partition(int[], int, int);  
29.void QuickSort(PHANSO[], int);
```

# Quick sort khử đệ quy



```
30.void HoanVi(PHANSO& a, PHANSO& b)
31.{
32.    PHANSO temp = a;
33.    a = b;
34.    b = temp;
35.}
```

# Quick sort khử đệ quy



```
36.int SoSanh(PHANSO x, PHANSO y)
37.{
38.    float a = (float)x.tu / x.mau;
39.    float b = (float)y.tu / y.mau;
40.    if (a > b)
41.        return 1;
42.    if (a < b)
43.        return -1;
44.    return 0;
45.}
```

```
46. int Partition(PHANSO a[], int Left, int Right)
47. {
48.     PHANSO pivot = a[Right];
49.     int vt = (Left - 1);
50.     for (int i=Left; i<=Right-1; i++)
51.         if (SoSanh(a[i], pivot)==-1)
52.         {
53.             vt++;
54.             HoanVi(a[vt], a[i]);
55.         }
56.     vt = vt + 1;
57.     HoanVi(a[vt], a[Right]);
58.     return vt;
59. }
```





# Quick sort khử đệ quy



```
60. void QuickSort(PHANSO a[], int n)
61. {
62.     if (n <= 1)
63.         return;
64.     QUEUE q;
65.     Init(q);

66.     SEGMENT ss = { 0, n - 1 };
67.     EnQueue(stk, ss);
68.     ...
```

# Quick sort khử đệ quy



```
69. | while (IsEmpty(q) == 0)
70. | {
71. |     ss = DeQueue(q);
72. |     int vt = Partition(a, ss.Left, ss.Right);
73. |     if (ss.Left < vt - 1)
74. |         EnQueue(q, { ss.Left, vt - 1 });
75. |     if (vt + 1 < ss.Right)
76. |         EnQueue(q, { vt + 1, ss.Right });
77. | }
78. }
```



**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**Hồ Thái Ngọc**

**ThS. Võ Duy Nguyên**

**TS. Nguyễn Tấn Trần Minh Khang**