



# QUICK SORT – STACK

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 Stack.



# 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 KHỦY 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



```
1. struct segment
2. {
3.     int Left;
4.     int Right;
5. };
6. typedef struct segment SEGMENT;
```

# Quick sort khử đệ quy



```
1. struct stack
2. {
3.     int n;
4.     SEGMENT a[10000];
5. };
6. typedef struct stack STACK;
```

# Quick sort khử đệ quy



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

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

9. `void` Init(STACK& );

10. `int` IsEmpty(STACK );

11. `int` IsFull(STACK );

12. `void` Push(STACK& , SEGMENT);

13. SEGMENT Pop(STACK& );

# Quick sort khử đệ quy



```
14.void HoanVi(PHANSO&, PHANSO&);  
15.int SoSanh(PHANSO, PHANSO);  
16.int Partition(PHANSO [], int, int);  
17.void QuickSort(PHANSO [], int);
```

# Quick sort khử đệ quy



```
18. void HoanVi(PHANSO& a, PHANSO& b)
19. {
20.     PHANSO temp = a;
21.     a = b;
22.     b = temp;
23. }
```

# Quick sort khử đệ quy



```
24.int SoSanh(PHANSO x, PHANSO y)
25.{
26.    float a = (float)x.tu / x.mau;
27.    float b = (float)y.tu / y.mau;
28.    if (a > b)
29.        return 1;
30.    if (a < b)
31.        return -1;
32.    return 0;
33.}
```

```
34. int Partition(PHANSO a[], int Left, int Right)
35. {
36.     PHANSO pivot = a[Right];
37.     int vt = (Left - 1);
38.     for (int i=Left; i<=Right-1; i++)
39.         if (SoSanh(a[i], pivot)==-1)
40.         {
41.             vt++;
42.             HoanVi(a[vt], a[i]);
43.         }
44.     vt = vt + 1;
45.     HoanVi(a[vt], a[Right]);
46.     return vt;
47. }
```





# Quick sort khử đệ quy



— Khái niệm: Khởi tạo ngăn xếp là tạo ra ngăn xếp rỗng không chứa đối tượng nào hết.

— Định nghĩa hàm

```
1. void Init(STACK& st)
2. {
3.     |     st.n = 0;
4. }
```

# Quick sort khử đệ quy



— Khái niệm: Kiểm tra ngăn xếp rỗng là hàm trả về giá trị 1 khi ngăn xếp rỗng. Trong tình huống ngăn xếp chưa rỗng thì hàm sẽ trả về giá trị 0.

— Định nghĩa hàm

```
1. int IsEmpty(STACK st)
2. {
3.     if(st.n==0)
4.         return 1;
5.     return 0;
6. }
```



# Quick sort khử đệ quy

— Khái niệm: Kiểm tra ngăn xếp đầy là hàm trả về giá trị 1 khi ngăn xếp đã đầy. Trong trường hợp ngăn xếp chưa đầy thì hàm trả về giá trị 0.

— Định nghĩa hàm

```
1. int IsFull(STACK st)
2. {
3.     if(st.n==1000)
4.         return 1;
5.     return 0;
6. }
```

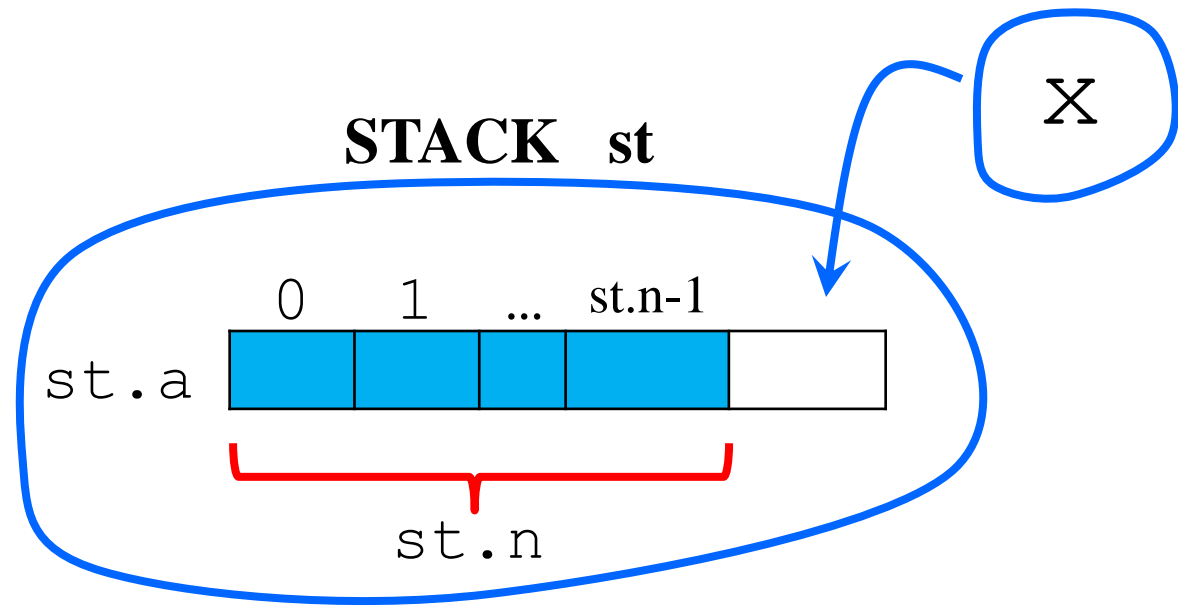
# Quick sort khử đệ quy



— Định nghĩa hàm

```
1. void Push(STACK& st, SEGMENT x)
2. {
3.   |   st.a[st.n] = x;
4.   |   st.n++;
5. }
```

— Hình vẽ minh họa



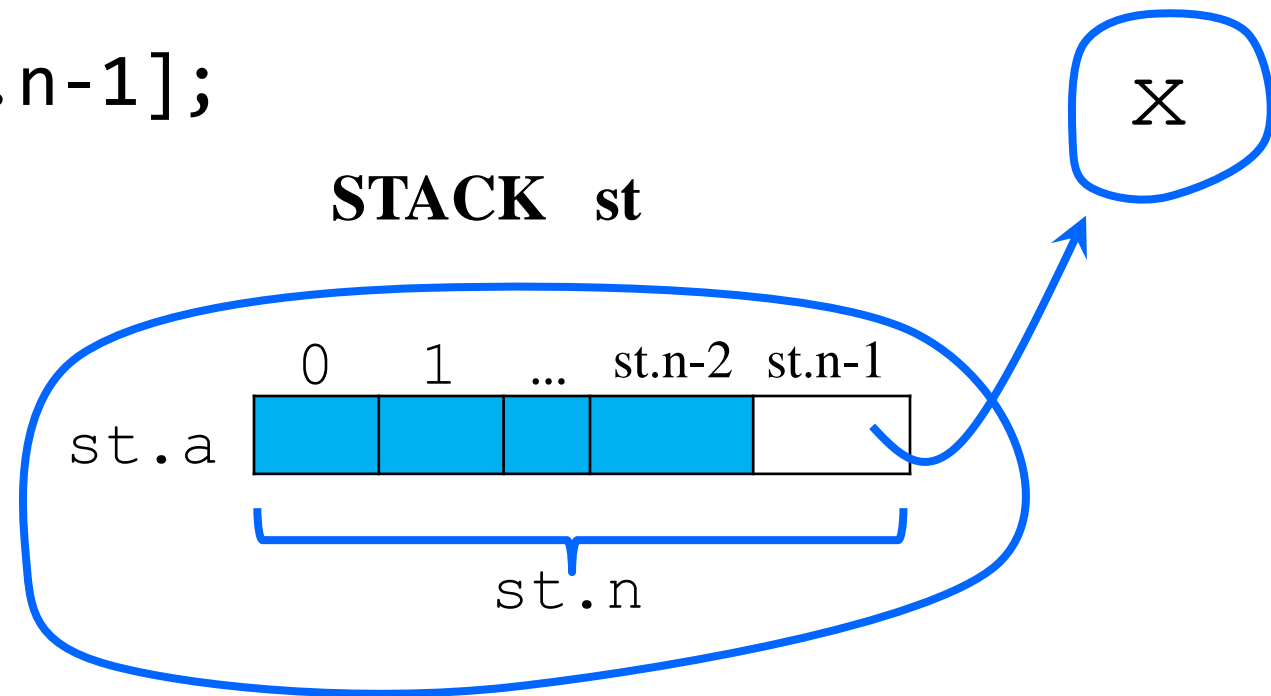
# Quick sort khử đệ quy



## — Định nghĩa hàm

```
1. SEGMENT Pop(STACK& st)
2. {
3.   KDL x = st.a[st.n-1];
4.   st.n--;
5.   return x;
6. }
```

## — Hình vẽ minh họa



# Quick sort khử đệ quy



```
48. void QuickSort(PHANSO a[], int n)
49. {
50.     if (n <= 1)
51.         return;
52.     STACK stk;
53.     Init(stk);

54.     SEGMENT dd = { 0, n - 1 };
55.     Push(stk, dd);
56.     ...
```

# Quick sort khử đệ quy



```
57. while (IsEmpty(stk) == 0)
58. {
59.     dd = Pop(stk);
60.     int vt = Partition(a, dd.Left, dd.Right);
61.     if (dd.Left < vt - 1)
62.         Push(stk, { dd.Left, vt - 1 });
63.     if (vt + 1 < dd.Right)
64.         Push(stk, { vt + 1, dd.Right });
65. }
66. }
```



**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**Hồ Thái Ngọc**

**ThS. Võ Duy Nguyên**

**TS. Nguyễn Tấn Trần Minh Khang**