Cấu trúc dữ liệu mảng:

Tìm kiếm:

1. Tìm kiếm tuần tự (n^2)

2.Tìm kiếm nhị phân (LOG2(n))

**SẮP XẾP DỮ LIỆU**

1.Tại sao cần sắp xếp dữ liệu

2.Sắp xếp dữ liệu căn bản

a) Ưu điểm:

i) dễ nhớ

ii) dễ cài đặt

b) Nhược điểm

i) chạy chậm

c) Các thuật toán

i) interchangeSort

ii) selectionSort (nhanh hơn interchangeSort 1 tí)

iii) InsertionSort (nhanh hơn nhiều so với 2 cái đầu)

iv) BubbleSort

v) ShackerSort

3.Sắp xếp dữ liệu nâng cao

a) Ưu điểm: tốc độ sắp xếp nhanh

b) Nhược điểm: cài đặt hơi phức tạp

c) Các thuật toán:

i) HeapSort

ii) ShellSort

iii) QuickSort

4.Ứng dụng

a) Sắp hạng của học sinh trong lớp.

b) Bảng xếp hạng của 1 giải bóng đá.

i) Điểm

ii) Hiệu số

iii) Số bàn thắng

iv) Đối đầu

v) Chỉ số fair play (số thẻ vàng và thẻ đỏ)

vi) Bốc thăm

Dô bài:

**InterchangeSort**

Input: 1 mảng có n phần tử

Output: mảng được sắp xếp tăng dần.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n = 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| i | J = i+1 | 7 | 5 | 3 | 9 | 4 | 6 | 8 | 2 | 1 |
| 0 | 1 | 5 | 7 | 3 | 9 | 4 | 6 | 8 | 2 | 1 |
|  | 2 | 3 | 7 | 5 | 9 | 4 | 6 | 8 | 2 | 1 |
|  | 3 | 3 | 7 | 5 | 9 | 4 | 6 | 8 | 2 | 1 |
|  | … |  |  |  |  |  |  |  |  |  |
|  | 6 | 3 | 7 | 5 | 9 | 4 | 6 | 8 | 2 | 1 |
|  | 7 | 2 | 7 | 5 | 9 | 4 | 6 | 8 | 3 | 1 |
|  | 8 | 1 | 7 | 5 | 9 | 4 | 6 | 8 | 3 | 2 |
| 1 | 2 | 1 | 5 | 7 | 9 | 4 | 6 | 8 | 3 | 2 |
|  | 3 | 1 | 5 | 7 | 9 | 4 | 6 | 8 | 3 | 2 |
|  | 4 | 1 | 4 | 7 | 9 | 5 | 6 | 8 | 3 | 2 |
|  | … |  |  |  |  |  |  |  |  |  |
|  | 7 | 1 | 3 | 7 | 9 | 5 | 6 | 8 | 4 | 2 |
|  | 8 | 1 | 2 | 7 | 9 | 5 | 6 | 8 | 4 | 3 |
| 2 | 3 | 1 | 2 | 7 | 9 | 5 | 6 | 8 | 4 | 3 |
|  | 4 | 1 | 2 | 5 | 9 | 7 | 6 | 8 | 4 | 3 |
|  | … |  |  |  |  |  |  |  |  |  |
|  | 7 | 1 | 2 | 4 | 9 | 7 | 6 | 8 | 5 | 3 |
|  | 8 | 1 | 2 | 3 | 9 | 7 | 6 | 8 | 5 | 4 |
|  |  |  |  |  |  |  |  |  |  |  |

**Source Code**

**void interchangeSoft (int A[], int n)**

**{**

**for (int i=0;i<n-1;i++)**

**{**

**for (int j=i+1;j<n;j++)**

**{**

**if (A[i] > A[j])**

**{**

**Swap(A[i],A[j]);**

**}**

**}**

**}**

**}**

**SELECTIONSORT**

**Source code**

**Void selectionSort(int A[], int n)**

**{**

**int temp, min\_index;**

**for (int i=0;i<n-1;i++)**

**{**

**min\_index = i;**

**//Tim phan tu nho nhat trong doan ben phai i**

**for (int j=i+1;j<n;j++)**

**{**

**if (A[i] > A[j])**

**{**

**min\_index = j;**

**}**

**}**

**//Hoan vi phan tu nho nhat voi i**

**if (i != min\_index)**

**{**

**temp = A[i];**

**A[i] = A[min\_index];**

**A[min\_index] = temp;**

**}**

**}**

**}**

**// Nguyen Phong Phu – 20110696**

**HEAPSORT**

**1. Định nghĩa HEAP:**

**Heap là 1 dãy số có n phần tử từ A0-An-1 sao cho với mọi I, ta luôn có**

**Ai >= A2i và Ai >= A2i+1**

**2. Tính chất của dãy HEAP:**

**a. A0 luôn là phần tử lớn nhất của dãy:**

**b. nếu xét 1 dãy con từ Ai - Aj mà 2i>j thì dãy nãy luôn thỏa mãn tính chất**

**của Heap**

**c. nếu 1 dãy thỏa mãn tính chất của Heap thì các dãy con của nó cũng thỏa mãn tính chất của Heap**

**3. Ví dụ**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **9** | **8** | **6** | **7** | **4** | **5** | **3** | **2** | **1** |
| **1** | **8** | **6** | **7** | **4** | **5** | **3** | **2** | **9** |
| **8** | **1** | **6** | **7** | **4** | **5** | **3** | **2** | **9** |

**4. Sắp xếp 1 dãy ngẫu nhiên thành 1 dãy Heap**

**5.**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **j** | **Max** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **4** | **4** |  | **5** | **1** | **8** | **4** | **3** | **6** | **9** | **7** | **2** |
| **3** | **3** | **6** | **5** | **1** | **8** | **9** | **3** | **6** | **4** | **7** | **2** |
|  | **6** |  |  |  |  |  |  |  |  |  |  |
| **2** | **2** | **5** | **5** | **1** | **8** | **9** | **3** | **6** | **4** | **7** | **2** |
| **1** | **1** | **3** | **5** | **9** | **8** | **1** | **3** | **6** | **4** | **7** | **2** |
|  | **3** | **7** | **5** | **9** | **8** | **7** | **3** | **6** | **4** | **1** | **2** |
| **0** | **1** | **1** | **9** | **5** | **8** | **7** | **3** | **6** | **4** | **1** | **2** |
|  | **1** | **2** | **9** | **8** | **5** | **7** | **3** | **6** | **4** | **1** | **2** |
|  | **2** | **5** | **9** | **8** | **6** | **7** | **3** | **5** | **4** | **1** | **2** |
|  | **5** |  | **9** | **8** | **6** | **7** | **3** | **5** | **4** | **1** | **2** |

**Nguyễn Phong Phú – 20110696**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **j** | **Max** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **6** | **0** | **1** | **7** | **1** | **6** | **4** | **3** | **5** | **2** | **8** | **9** |
|  | **1** | **2** | **7** | **6** | 1 | **4** | **3** | **5** | **2** | **8** | **9** |
|  | **2** | **5** | **7** | **6** | **5** | **4** | **3** | **1** | **2** | **8** | **9** |
|  | **5** |  |  |  |  |  |  |  |  |  |  |

**QUICKSORT**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **l** | **r** | **i** | **j** | **x** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| **0** | **12** | **0** | **12** | **5** | **7** | **2** | **5** | **4** | **6** | **3** | **5** | **5** | **2** | **4** | **7** | **5** | **3** |
|  |  | **0** | **12** |  | **3** |  |  |  |  |  |  |  |  |  |  |  | **7** |
|  |  | **2** | **11** |  |  |  | **5** |  |  |  |  |  |  |  |  | **5** |  |
|  |  | **4** | **9** |  |  |  |  |  | **4** |  |  |  |  | **6** |  |  |  |
|  |  | **6** | **8** |  |  |  |  |  |  |  | **2** |  | **5** |  |  |  |  |
|  |  | **7** | **7** |  | **3** | **2** | **5** | **4** | **4** | **3** | **2** | **5** | **5** | **6** | **7** | **5** | **7** |
|  |  | **8** | **6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **0** | **6** | **0** | **6** | **4** | **3** | **2** | **5** | **4** | **4** | **3** | **2** |  |  |  |  |  |  |
|  |  | **2** | **6** |  |  |  | **2** |  |  |  | **5** |  |  |  |  |  |  |
|  |  | **3** | **5** |  |  |  |  | **3** |  | **4** |  |  |  |  |  |  |  |
|  |  | **4** | **4** |  |  |  |  |  | **4** |  |  |  |  |  |  |  |  |
|  |  | **5** | **3** |  | **3** | **2** | **2** | **3** | **4** | **4** | **5** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Nguyễn Phong Phú – 20110696**

**SHELLSORT**

**B1 dùng vòng lặp để lấy step;**

**N = 7;**

**Step = 1**

**N = 3**

do

{

step = step \* 2 + 1; step = 3

} while (2 \* step + 1 < n);

Step = (step – 1)/2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Step** | **i** | **j** |  | **0** | **1** | **2** | **3** | **4** | **5** | **6** |
| **3** | **3** | **0** |  | **7** | **4** | **5** | **3** | **1** | **2** | **6** |
|  |  | **-3** |  | **7** |  |  | **7** |  |  |  |
|  |  |  |  | **3** |  |  | **7** |  |  |  |
|  | **4** | **1** |  |  | **4** |  |  | **4** |  |  |
|  |  | **-2** |  |  | **1** |  |  |  |  |  |
| **1** | **1** | **0** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

**BÀI TẬP ỨNG DỤNG**Text

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application, Word

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Diagram

Description automatically generated