Bài 1 : Cho dãy : 50 8 34 6 98 17 83 25 66 42 21 59 63 71 85

* **Sắp xếp vun đống** (heapsort)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 50 | 8 | 34 | 6 | 98 | 17 | 83 | 25 | 66 | 42 | 21 | 59 | 63 | 71 | 85 |

*Cây nhị phân biểu diễn dãy khóa*

+ Cây con có nút gốc là 83 (vị trí thứ 7 – X[6] trở thành đống)

*Xử lý nút khóa X[6]*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 50 | 8 | 34 | 6 | 98 | 17 | 85 | 25 | 66 | 42 | 21 | 59 | 63 | 71 | 83 |

+ Cây con có nút gốc 17 trở thành đống

*Xử lý nút khóa X[5]*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 50 | 8 | 34 | 6 | 98 | 63 | 85 | 25 | 66 | 42 | 21 | 59 | 17 | 71 | 83 |

+ Câ con có nút gốc 98 trở thành đống

*Xử lý nút khóa X[4]*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 50 | 8 | 34 | 6 | 98 | 63 | 85 | 25 | 66 | 42 | 21 | 59 | 17 | 71 | 83 |

+Cây con có nút gốc 6 trở thành đống

*Xử lý nút khóa X[3]*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 50 | 8 | 34 | 66 | 98 | 63 | 85 | 25 | 6 | 42 | 21 | 59 | 17 | 71 | 83 |

+Cây con có nút gốc 34 trở thành đống

*Xử lý nút khóa X[2]*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 50 | 8 | 85 | 66 | 98 | 63 | 83 | 25 | 6 | 42 | 21 | 59 | 17 | 71 | 34 |

**+** Cây con có nút gốc 8 trở thành đống

*Xử lý nút khóa X[1]*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 50 | 98 | 85 | 66 | 42 | 63 | 83 | 25 | 6 | 8 | 21 | 59 | 17 | 71 | 34 |

+ Cây có nút gốc 50 trở thành đống

*Xử lý nút khóa X[0]*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 98 | 66 | 85 | 50 | 42 | 63 | 83 | 25 | 6 | 8 | 21 | 59 | 17 | 71 | 34 |

+ Khóa trội 98 chuyển đến vị trí cuối cùng – hoán vị cho khóa 34, nút ứng với khóa trội 98 coi như được loại khỏi cây

*Hoán đổi nút đầu với nút cuối*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 34 | 66 | 85 | 50 | 42 | 63 | 83 | 25 | 6 | 8 | 21 | 59 | 17 | 71 | **98** |

+Cây còn lại vun lại thành đống

+ Xét nút gốc của cây ta được

*Đống thứ 2*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 85 | 66 | 85 | 50 | 42 | 63 | 71 | 25 | 6 | 8 | 21 | 59 | 17 | 34 | **98** |

Lại tiếp tục đổi vị trí đầu tiên và vị trí cuối cùng cuối cùng dãy sẽ được sắp xếp

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
| 6 | 8 | 17 | 21 | 25 | 34 | 42 | 50 | 59 | 63 | 66 | 71 | 83 | 85 | **98** |

#include<iostream>

using namespace std;

void NhapDanhSach(int a[], int n)

{

cout << "\t\tNhap danh sach " << endl;

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

{

cout << "Nhap vi tri " << i + 1<<" :";

cin >> a[i];

}

}

void Xuat(int a[], int n)

{

cout << "\t\t Danh sach " << endl;

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

{

cout << a[i]<<"\t";

}

cout << endl;

}

void HoanVi(int a[], int d, int n)

{

if (d <= n / 2 - 1)

{

int i = 2\*d + 1;

if (a[i] < a[i + 1] && i + 1 < n)

i++;

if (a[i] > a[d])

{

int p = a[i];

a[i] = a[d];

a[d] = p;

HoanVi(a, i, n);

}

}

}

void TaoDongDau(int a[], int n)

{

for (int i = n / 2 - 1; i >= 0; i--)

{

HoanVi(a, i, n);

}

}

void SapXep(int a[], int n)

{

TaoDongDau(a, n);

for (int i = n; i > 1; i--)

{

int p = a[0];

a[0] = a[i - 1];

a[i - 1] = p;

HoanVi(a, 0, i - 1);

}

}

int main()

{

int n;

cout << "Nhap n:";

cin >> n;

int\* a = new int[n];

NhapDanhSach(a, n);

cout << "\t\t Truoc khi sap xep" << endl;

Xuat(a, n);

SapXep(a, n);

cout << "\t\t Khi sap xep" << endl;

Xuat(a, n);

system("pause");

return 0;

}

**- Sắp xếp Phân Đoạn**

Chọn t=25 làm chốt, i=Left=0, j=Right=14

i

j

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 50 | 8 | 34 | 6 | 98 | 17 | 83 | 25 | 66 | 42 | 21 | 59 | 63 | 71 | 85 |

Tăng i cho đến khi gặp khóa không nhỏ hơn chốt.

j

j

i

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 50 | 8 | 34 | 6 | 98 | 17 | 83 | 25 | 66 | 42 | 21 | 59 | 63 | 71 | 85 |

Giảm j cho đến khi gặp khóa không lớn hơn chốt

i

j

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 34 | 6 | 98 | 17 | 83 | 25 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Đổi chỗ Xi và Xj (khi i<j)

i

j

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 34 | 6 | 98 | 17 | 83 | 25 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Tăng i thêm 1 đơn vị và giảm j đi 1 đơn vị, tiếp tục quá trình

i

i

j

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 34 | 6 | 98 | 17 | 83 | 25 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Tăng i cho đến khi gặp khóa không nhỏ hơn chốt

j

j

i

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 34 | 6 | 98 | 17 | 83 | 25 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Giảm j cho đến khi gặp khóa không lớn hơn chốt

j

i

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 25 | 6 | 98 | 17 | 83 | 34 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Đổi chỗ i và j ( khi i<j)

j

i

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 25 | 6 | 98 | 17 | 83 | 34 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Tăng i lên 1 đơn vị và giảm j xuống 1 đơn vị

i

i

j

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 25 | 6 | 98 | 17 | 83 | 34 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Tăng i cho đến khi gặp khóa không nhỏ hơn chốt

j

j

i

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 25 | 6 | 98 | 17 | 83 | 34 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Giảm j cho đến khi gặp khóa không lớn hơn chốt

j

i

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 25 | 6 | 17 | 98 | 83 | 34 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Đổi chỗ i và j ( khi i<j)

i

j

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 8 | 25 | 6 | 17 | 98 | 83 | 34 | 66 | 42 | 50 | 59 | 63 | 71 | 85 |

Tăng i lên 1 đơn vị và giảm j xuống 1 đơn vị

Đến đây kết thúc sắp xếp lượt 1 vì i > j.

Sau một lượt duyệt dãy khóa chia làm 2 phân đoạn, đoạn 1 từ Xleft  đến Xj  gồm các phần tử không lớn hơn chốt và đoạn 2 từ Xi  đến Xright  gồm các phần từ không nhỏ hơn chốt.

Tiếp tục sắp xếp đoạn 1 và đoạn 2 theo kĩ thuật như trên. Quá trình phân đoạn kết thúc khi chỉ còn 1 phần tử.

Đến đây hoàn thành việc sắp xếp.

#include<iostream>

using namespace std;

void NhapDanhSach(int a[], int n)

{

cout << "\t\tNhap danh sach " << endl;

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

{

cout << "Nhap vi tri " << i + 1 << " :";

cin >> a[i];

}

}

void Xuat(int a[], int n)

{

cout << "\t\t Danh sach " << endl;

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

{

cout << a[i] << "\t";

}

cout << endl;

}

void QuickSort(int a[], int Left, int Right)

{

if (Left < Right)

{

int i = Left;

int j = Right;

int t = a[(Left + Right) / 2];

while (i <= j)

{

while (a[i] < t) i++;

while (a[j] > t) j--;

if (i <= j)

{

int p = a[i];

a[i] = a[j];

a[j] = p;

i++;

j--;

}

}

QuickSort(a, Left, j);

QuickSort(a, i, Right);

}

}

int main()

{

int n;

cout << "Nhap n:";

cin >> n;

int\* a = new int[n];

NhapDanhSach(a, n);

cout << "\t\t Truoc khi sap xep" << endl;

Xuat(a, n);

QuickSort(a, 0, n - 1);

cout << "\t\t Khi sap xep" << endl;

Xuat(a, n);

system("pause");

return 0;

}

- **Phương pháp sắp xếp trộn**

Coi dãy khóa gồm 15 vệt có độ dài 1

[50] [ 8 ] [ 34 ] [ 6 ] [ 98 ] [ 17 ] [ 83 ] [ 25 ] [ 66 ] [ 42 ] [ 21 ] [ 59 ] [ 63 ] [71 ] [ 85 ]

Trộn các vệt có độ dài 1 kề nhau được

[ 8 50 ] [ 6 34 ] [17 98 ] [ 25 83 ] [ 42 66 ] [21 59 ] [63 71 ] [85]

Giữ nguyên cặp lẻ [85]

Trộn các vệt kề nhau ta được

[ 6 8 34 50] [ 17 25 83 98 ] [ 21 42 59 66 ] [ 63 71 85]

Trộn các cặp vệt kề nhau ta được

[ 6 8 17 25 34 50 83 98 ] [ 21 42 59 63 66 71 85]

Trộn hai vệt cuối cùng ta được

[ 6 8 17 21 25 34 42 50 59 63 66 71 83 85 98 ]

Như vậy ta được dãy gồm các phần tử theo thứ tự tăng dần

Bài 1D: Minh họa phương pháp tìm kiếm tuần tự khóa k1=98, k2=63, k3=44 trên dãy khóa X và tính số phép toán so sánh trong mỗi trường hợp

50 8 34 6 98 17 83 25 66 42 21 59 63 71 85

**Minh họa phép tìm kiếm tuần tự khóa k1 = 98**

Với i=0<n-1 Ro.key=50 ≠ k1 chuyển sang so sánh bản ghi tiếp theo

Với i=1<n-1 Ro.key=8 ≠ k1 chuyển sang so sánh bản ghi tiếp theo

Với i=2<n-1 Ro.key=34 ≠ k1 chuyển sang so sánh bản ghi tiếp theo

Với i=3<n-1 Ro.key=6 ≠ k1 chuyển sang so sánh bản ghi tiếp theo

Với i=4<n-1 Ro.key=98 = k1 kết thúc phần tìm kiếm và trả về vị trí tìm được i=4.

**Minh họa phép tìm kiếm tuần tự khóa k2 = 63**

Với i=0<n-1 Ro.key=50 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=1<n-1 Ro.key=8 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=2<n-1 Ro.key=34 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=3<n-1 Ro.key=6 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=4<n-1 Ro.key=98 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=5<n-1 Ro.key=17 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=6<n-1 Ro.key=83 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=7<n-1 Ro.key=25 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=8<n-1 Ro.key=66 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=9<n-1 Ro.key=42 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=10<n-1 Ro.key=21 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=11<n-1 Ro.key=59 ≠ k2 chuyển sang so sánh bản ghi tiếp theo

Với i=12<n-1 Ro.key=63 = k2 kết thúc phần tìm kiếm và trả về vị trí tìm được i=12.

**Minh họa phép tìm kiếm tuần tự khóa k3 = 44**

Với i=0<n-1 Ro.key=50 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với si=0<n-1 Ro.key=50 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=1<n-1 Ro.key=8 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=2<n-1 Ro.key=34 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=3<n-1 Ro.key=6 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=4<n-1 Ro.key=98 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=5<n-1 Ro.key=17 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=6<n-1 Ro.key=83 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=7<n-1 Ro.key=25 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=8<n-1 Ro.key=66 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=9<n-1 Ro.key=42 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=10<n-1 Ro.key=21 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=11<n-1 Ro.key=59 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=12<n-1 Ro.key=63 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=13<n-1 Ro.key=71 ≠ k3 chuyển sang so sánh bản ghi tiếp theo

Với i=14 = n-1 Ro.key=85 ≠ k3

=> kết thúc, không tìm được vị trí i.

**Câu 1.(3 điểm)**

Cho dãy số nguyên X = { 8, 21, 7, 56, 10, -35, 19}

*Yêu cầu:* - Minh họa việc sắp xếp dãy X theo chiều giảm dần bằng phương pháp nổi bọt.

-Viết giải thuật sắp xếp dãy X theo chiều giảm dần bằng phương pháp nổi bọt.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lần  duyệt | X0 | X1 | X2 | X3 | X4 | X5 | X6 | Giải thích |
| 8 | 21 | 7 | 56 | 10 | -35 | 19 |
| i = 0 | 8 | 21 | 7 | 56 | 10 | **19** | **-35** | So sánh X0 với X5 và đổi chỗ |
| 8 | 21 | 7 | 56 | **19** | **10** | -35 | So sánh X5 với X4 và đổi chỗ |
| 8 | 21 | 7 | **56** | **19** | 10 | -35 | So sánh X4 với X3 và giữ nguyên |
| 8 | 21 | **56** | **7** | 19 | 10 | -35 | So sánh X3 với X2 và đổi chỗ |
| 8 | **56** | **21** | 7 | 19 | 10 | -35 | So sánh X2 với X1 và đổi chỗ |
| **56** | **8** | 21 | 7 | 19 | 10 | -35 | So sánh X1 với X0 và đổi chỗ |
| i = 1 | **56** | **21** | 8 | 19 | 7 | 10 | -35 | Tương tự lần duyệt i = 0 |
| i = 2 | 56 | 21 | **19** | 8 | 10 | 7 | -35 |  |
| i = 3 | 56 | 21 | 19 | **10** | 8 | 7 | -35 |  |
| i = 4 | 56 | 21 | 19 | 10 | **8** | 7 | -35 |  |
| i = 5 | 56 | 21 | 19 | 10 | 8 | **7** | -35 | Dãy được sắp xếp |

- Giải thuật

void BubleSort(int a[], int n)

{

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

{

for (int j = n - 1; j > i; j--)

{

if (a[j] > a[j - 1])

{

int p = a[j];

a[j] = a[j - 1];

a[j - 1] = p;

}

}

}

}

**Câu 2.(3 điểm)**Cho dãy số nguyên X = {15, 20, 7, 56, 10, -35, 19}

*Yêu cầu:*

- Minh họa việc sắp xếp dãy X theo chiều tăng dần bằng phương pháp lựa chọn.

-Viết giải thuật sắp xếp dãy X theo chiều tăng dần bằng phương pháp lựa chọn.

Bài Làm

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lần  duyệt | X0 | X1 | X2 | X3 | X4 | X5 | X6 | Giải thích |
| 15 | 20 | 7 | **56** | 10 | -35 | 19 |
| i = 0 | 56 | 20 | 7 | 15 | 10 | -35 | 19 | Duyệt từ X0 đến X6 , X3 lớn nhât đổi chỗ cho X0 |
| i = 1 | 56 | 20 | 7 | 15 | 10 | -35 | **19** | Duyệt từ X1 đến X6 , X1 lớn nhât đổi giữ nguyên |
| i = 2 | 56 | 20 | 19 | 15 | 10 | -35 | 7 | Duyệt từ X2 đến X6 , X6 lớn nhât đổi chỗ cho X2 |
| i = 3 | 56 | 20 | 19 | 15 | 10 | -35 | 7 | Duyệt từ X3 đến X6 , X3 lớn nhât giữ nguyên |
| i = 4 | 56 | 20 | 19 | 15 | 10 | -35 | 7 | Duyệt từ X4 đến X6 , X4 lớn nhất giữ nguyên |
| i = 5 | 56 | 20 | 19 | 15 | 10 | 7 | -35 | Duyệt từ X5 đến X6 , X6 lớn nhât đổi chỗ cho X5 |

- Giải thuật

void SelectionSort(int a[], int n)

{

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

{

int m = i;

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

{

if (a[m] > a[j])

m = j;

}

if (m != i)

{

int p = a[i];

a[i] = a[m];

a[m] = p;

}

}

}

**Câu 3.(3 điểm)**

Cho dãy số nguyên X = {15, 20, 7, 56, 10, -35, 19}

*Yêu cầu:*

- Minh họa việc sắp xếp dãy X theo chiều tăng dần bằng phương pháp chèn.

- Viết giải thuật sắp xếp dãy X theo chiều tăng dần bằng phương pháp chèn.

Bài làm

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lần  duyệt | X0 | X1 | X2 | X3 | X4 | X5 | X6 | Giải thích |
| 15 | 20 | 7 | 56 | 10 | **-**35 | 19 |
| i = 0 | 20 | **15** | 7 | 56 | 10 | **-**35 | 19 | Lấy t = X1 = 20 chèn vào dãy đích |
| i = 1 | 20 | 15 | **7** | 56 | 10 | **-**35 | 19 | Lấy t = X2 = 7 chèn vào dãy đích |
| i = 2 | 20 | 15 | 7 | **56** | 10 | -35 | 19 | Lấy t = X3 = 56 chèn vào dãy đích |
|  | 20 | 20 | 15 | 7 | 10 | -35 | 19 |  |
| i = 3 | 56 | 20 | 15 | 7 | **10** | -35 | 19 | Lấy t = X4 =10 chèn vào dãy đích |
|  | 56 | 20 | 15 | 7 | 7 | -35 | 19 |  |
| i = 4 | 56 | 20 | 15 | 10 | 7 | **-35** | 19 | Lấy t = X5 = -35 chèn vào dãy đích |
| i = 5 | 56 | 20 | 15 | 10 | 7 | -35 | 19 | Lấy t = X6 = 19 chèn vào dãy đích |
|  | 56 | 20 | 15 | 15 | 10 | 7 | -35 |  |
|  | 56 | 20 | 19 | 15 | 10 | 7 | -35 | Dãy được sắp xếp |

- Giải thuật

void InsertSort(int a[], int n)

{

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

{

int t = a[i];

int j = i - 1;

while (j >= 0 && t > a[j])

{

a[j + 1] = a[j];

j--;

}

a[j + 1] = t;

}

}