**Câu 1.**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 = 8 | X1 = 21 | X2 = 7 | X3 = 56 | X4 = 10 | X5 = -35 | X6 = 19 | Giải thích |
| i=0 | 8 | 21 | 7 | 56 | 10 | **19** | **-35** | So sánh cặp X5,X6  và đổi chỗ |
| 8 | 21 | 7 | 56 | **19** | **10** | -35 | So sánh cặp X4,X5  và đổi chỗ |
| 8 | 21 | 7 | **56** | **19** | 10 | -35 | So sánh cặp X3,X4  và giữ nguyên |
| 8 | 21 | **56** | **7** | 19 | 10 | -35 | So sánh cặp X2,X3  và đổi chỗ |
| 8 | **56** | **21** | 7 | 19 | 10 | -35 | So sánh cặp X1,X2  và đổi chỗ |
| **56** | **8** | 21 | 7 | 19 | 10 | -35 | So sánh cặp X0,X1  và đổi chỗ |
| i=1 | 56 | **21** | 8 | 19 | 7 | 10 | -35 | Tương tự với 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 bubbleSort(int X[], int n){

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

    for(int j = n - 1; j > i; j--){

      if(X[j] > X[j-1]){

        int tg = X[j];

        X[j] = X[j-1];

        X[j-1] = tg;

      }

    }

  }

}

**Câu 2.**

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

*Yêu cầu:*

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

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

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

**Giải thuật:**

void selectSort(int X[], int n){

int m;

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

m = i;

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

if(X[j] > X[m]){

m = j;

}

}

if(m!=i){

int tg = X[i];

X[i] = X[m];

X[m] = tg;

}

}

}

**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 giảm dần bằng phương pháp chèn.

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

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

**Giải thuật:**

void insertSort(int X[], int n){

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

int tg = X[i];

int j = i -1;

while(j>=0 && X[j] < tg){

X[j+1]=X[j];

j--;

}

X[j+1]=tg;

}

}