1.1.1 Sắp xếp vun đống (HeapSort)

í tưởng:

Tương tự như sắp xếp lựa chọn đơn giản, tuy nhiên ta sử dụng cấu trúc dữ liệu theo cấu trúc cây.

Cấu trúc dữ liệu:

- Cây cân bằng, các lá nằm trên cùng mức của cây và nằm nhiều nhất ở hai mức liên tiếp nhau và các lá ở mức dưới ở những vị trí bên trái nhất.
- Giá trị dữ liệu ở mỗi nút lớn hơn giá trị của những mục ở các con của nó.

Ta sử dụng cấu trúc dữ liệu mảng để lưu trữ dữ liệu vào.

Giải thuật: Gồm hai công việc chính sau đây:

Từ mảng dữ liệu vào ta xây dựng cây thoả mãn 2 điều kiện nêu trên. Sau mỗi bước dựng cây, phần từ có giá trị lớn nhất sẽ nằm ở gốc của cây.

Ta tiến hành loại bỏ gốc này (thông qua việc hoán vị chúng với lá cuối cùng). Sau đó ta lại xây dựng cây mới (cây này đã được cắt bỏ lá cuối cùng).

Công việc trên tiến hành sau hữu hạn bước (vì số phần từ là hữu hạn). Dãy được sắp.

```
Procedure HeapSort(var X:HeapType; n :integer)
Begin
    {Xây dựng cây}
    HeapUp(X,n);
    {Đặt phần tử ở gốc vào cuối danh  sách, lấy nó ra khỏi cây và thực hiện
    xây  dựng cây cho phần còn lại}
    For i:=n dowto 2 do
          Begin
                Permute(X(1),X(i)) { hoan vi x(1),x(i)}
                Heapdown(X,1,i-1);
          End;
End;
Procedure HeapUp (Var Heap: HeapType; n:integer)
    Var r: integer {chi so}
    Begin
          For r:=n div 2 downto 1 do
                HeapDown(Heap,r,n);

    End;
Procedure HeapDown (Var Heap:HeapType; r,n: integer)
{ Tim nut goc cho cay}
Var
    Child : integer; { con lon nhat}
    Done : Boolean; {Bien trang thai bao hieu hoan tat thu tuc}
Begin
    Done:=False;
```

```
    Child:= 2*r;
    While (not done) and (child <=n) do
          Begin
                If (Child <n) Then
                      If Heap(Child) < Heap(Child +1) then
                            Child :=Child +1;
                      If Heap(r)<Heap(child) then
                            Begin
                              Permute(Heap(r),Heap(Child));
                              R:=child
                              Child:=2*child;
                            End;
                      Else
                            Done:=true;

          End;
End;
```

Độ phức tạp : $O(n\log_2 n)$
So sánh: Khá tốt

1.1.2 Sắp xếp nhanh (QuickSort)

ý tưởng:

Dùng phương pháp chia để trị, sử dụng đệ quy. Trong mỗi bước của thuật toán , tiến hành lựa chọn phần từ sao cho các phần từ ở bên trái nó đều nhỏ hơn nó, các phần từ bên phải đều lớn hơn nó. Có nhiều phiên bản, trình bày phiên bản cơ sở.

Cấu trúc dữ liệu:
Cài đặt trên cơ sở mảng.

Giải thuật:

```
Procedure QuickSort(Var l,r:integer);
    Var pos: integer;
    Begin
          If l < r then
                Begin
                      Pos:= Split(l,r);
                      QuickSort(l,pos-1);
                      QuickSort(pos+1,r);
                End;
    End;
```

Nếu ta xây dựng được thuật toán split thì bài toán được giải. Sau đây ta tiến hành xây dựng thuật toán này. Trong thuật toán ta sẽ đi tìm vị trị pos (gọi là điểm chốt) sao cho x(pos) thoả điều kiện sau : các phần tử bên trái điểm chốt nhỏ hơn x(pos), các phần từ bên phải điểm chốt lớn hơn x(pos).

Ta chọn l là điểm chốt. Khi đó ta tiến hành:

Quét từ trái sang cho đến khi gặp phần tử a_i lớn hơn a_l.

Quét từ phải sang cho đến khi gặp phần tử a_j nhỏ hơn a_l.

Đổi chỗ a_i, a_j.

Lặp lại các bước trên hữu hạn lần (vì số phần từ là giới hạn) cho đến khi 2 con trỏ gặp nhau hoặc vượt sang biên và tiến hành hoán vi a_l với phần từ trái nhất của tập con bên phải.

```
Procedure Split(Var low,high :integer);
   Var left,right :integer;item:datatype;
   Begin
        Item:=x[low];
        Left:=Low;
        Right:=high;
        While left <right do
             Begin
                  While x[right]> item do
                     Right:=right -1;
                     While  (left  <right)  and  x[left]<=item)
do
                       Left:=left +1;
                       If (left <right) then
                            Permute (X[left],x[right])
                            Pos:=Right;
             End;
   End;
```

1.1.3 Sắp xếp trộn (MergeSort)

ý tưởng:

Ta coi dãy cần sắp bao gồm hai dãy con đã được sắp thứ tự, ta xây dựng dãy phần tử từ hai dãy trên sao cho dãy này cũng được sắp.

Giải thuật:

Có hai phương pháp trộn trực tiếp (trộn nhị phân) và trộn tự nhiên. Phương pháp trộn tự nhiên có ưu điểm là tận dụng được các đoạn đã được sắp thứ tự trong dãy được sắp.

Trong phương pháp trộn tự nhiên .Lặp lại các bước sau:

- Chia tệp F thành F1,F2 bằng cách sao luân phiên các tệp con có thứ tự tự nhiên vào F1 và F2
- Trộn các tệp con tương ứng trong F1 và F2 vào F

Hai công việc lặp lại cho đến khi NumSubFiles =1 . NumSubFiles là số các tệp con có thứ tự tạo ra trong F. Sau đây là thủ tục MergeSort

Giả sư các Tệp có cấu trúc như sau:

```
   Type f: File of Item;
   Var Z:f;
Procedure  MergeSort;
   Var X,Y:f;
       NumSubFiles :Integer;
   Begin
        Repeat
             Rewrite(X);Rewrite(Y);Reset(Z);
             { Chia doan tu Z vao X va Y}
             Distribute;
             Reset(X);Reset(Y);Rewrite(Z);
             NumSubFiles =0;
             Merge;
        Until NumSubFiles =1;
   End.
```

Ta nhận thấy thủ tục MergeSort sẽ hoàn thiện hay nói cách khác bài toán sẽ được giải nếu như ta xây dựng được hai thủ tục Distribute và Merge.

Ta xây dựng thủ tục Distribute :

Mở file F1 và F2 để ghi, F để đọc.

While (chưa đạt kết thúc F) làm các bước sau:

Sao một tệp con có thứ tự của F vào F1 như sau : thự hiện một cách lặp lại việc đọc một phần tử tiếp theo của F và viết nó vào F1 cho đến khi phần tử tiếp theo trong F nhỏ hơn phần tử được sao hay đạt kết thúc F.

Nếu chưa đạt kết thúc F, sao tệp con có thứ tự tiếp theo của F vào F2 theo cách tương tự.

```
Procedure Distribute;
   Begin
        Repeat
             CopyRun(z,x);
             If not Eof(z) then CopyRun(z,y);
        Until Eof(z);
```

Thủ tục CopyRun(a,b :f) thực hiện việc trộn đoạn có thứ tự tự nhiên từ a vào b. Thủ tục này tiến hành chép một đoạn có thứ tự tự nhiên từ a vào b cho đến khi kết thúc đoạn có thứ tự tự nhiên. Để đánh dấu việc kết thúc đoạn có thứ tự tự nhiên ta dùng biến logic eor (end of run). Nếu eor =True :kết thúc đường chạy. Sau đây là thủ tục CopyRun

```
Procedure CopyRun(Var a,b :f);
Begin
```

```
    Repeat
          Copy(a,b);
    Until eor;
End;
```

Thủ tục Copy(a,b :f) thực hiện sao chép từng phần tử trong a vào b cho đến khi đạt kết thúc a hay phần tử tiếp theo trong a nhỏ hơn phần từ được ghi vào b.

```
Procedure Copy(Var a,b:f);
    {bien buf :la mot phan tu trong a, gia su rang cac item co
truong key la khoa de sap}
    Var buf :item
    Begin
          Read(a,buf);Write(b,buf);
          If eof(a) then eor := true
          Else eor := buf.key > x^.key;
    End;
```

Tiếp theo ta xây dựng thủ tục Merge:

1.Mở file F1,F2 để đọc, mở file F để ghi.

2. While (chưa đạt kết thúc F1) và (chưa đạt kết thúc F2) do

a.While (chưa đạt kết thúc của tệp con trong F1) và (chưa đạt kết thúc trong tệp con trong F2) do

Nếu phần tử tiếp theo trong F1 nhỏ hơn phần tử tiếp theo trong F2 thì sao phần tử tiếp theo trong F1 vào F; ngược lại thì sao phần tử tiếp theo trong F2 vào F.

b.Nếu đạt kết thúc của một tệp con trong F1, sao phần tử còn lại của tệp con tương ứng trong F2 vào F; ngược lại sao phần còn lại của tệp con tương ứng trong F1 vào F.

c.Tăng NumSubFiles thêm 1

3.Sao tất cả các tệp con còn lại trong F1 hay F2 vào F, với mỗi tệp con tăng NumSubFiles thêm 1

Dưới đây la thủ tục Merge

```
Procedure Merge;
Begin
    While not eof (X) and not eof(Y) do
          Begin
                Repeat
                      If x^.key <= y^.key then
                            Begin
                                  Copy(x,z);
                                  If eor then CopyRun(y,z);
                            End
                      Else
                            Begin
                                  Copy(y,z);
                                  If oer then CopyRun(x,z);
```

```
                        End;
                Until eor;
                NumSubFiles =NumSubFiles +1;
        End;
        While not eof(X) do
                Begin
                        CopyRun(X,Z);
                        NumSubFiles =NumSubFiles +1;
                End;
        While not oef(Y) do
                Begin
                        CopyRun(Y,Z);
                        NumSubFiles =NumSubFiles +1;
                End;
End;
```

Độ phức tạp : $O(n\log_2 n)$

So sánh: Khá tốt