2.6 Dùng phương pháp xếp thứ tự HeapSort, mô tả từng bước quá trình xếp thứ tự dãy số trên (không lập trình). Tín độ phức tạp của quá trình xếp thứ tự danh sách trên.

Ta có danh sách không theo thứ tự

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 70 | 20 | 60 | 90 | 10 | 50 | 30 |

vì 90 > 30 và lớn hơn mọi số khác nên ưu tiên đưa về cuối mảng. Nên 90 đổi chỗ với 30.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 70 | 20 | 60 | 30 | 10 | 50 | 90 |

90 được loại bỏ ra khỏi heap vì nó đã được sắp xếp đúng vị trí

- heap mới được tạp và tiếp tục 70 đổi chỗ cho 50

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 50 | 20 | 60 | 30 | 10 | 70 | 90 |

70 được loại bỏ ra khỏi heap vì nó đã được sắp xếp đúng vị trí

- heap mới được tạp và tiếp tục 60 đổi chỗ cho 10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 50 | 20 | 10 | 30 | 60 | 70 | 90 |

60 được loại bỏ ra khỏi heap vì nó đã được sắp xếp đúng vị trí

- heap mới được tạp và tiếp tục 50 đổi chỗ cho 30

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 30 | 20 | 10 | 50 | 60 | 70 | 90 |

50 được loại bỏ ra khỏi heap vì nó đã được sắp xếp đúng vị trí

- heap mới được tạp và tiếp tục 40 đổi chỗ cho 10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 30 | 20 | 40 | 50 | 60 | 70 | 90 |

40 được loại bỏ ra khỏi heap vì nó đã được sắp xếp đúng vị trí

- heap mới được tạp và tiếp tục 30 đổi chỗ cho 20

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 90 |

Tất cả các phần tử đã sắp xếp đúng vị trí và dừng chương trình . Ta sẽ được kết quả các giá trị phần tử sắp xếp với giá trị tăng dần

Đánh giá thuật toán:

Giải thuật heap sort gồm hai giai đoạn sau:

§ Giai đoạn 1: Hiệu chỉnh dãy số ban đầu thành Heap

§ Giai đoạn 2: Sắp xếp dãy số dựa trên Heap

o Bước 1:Đưa phần tử lớn nhất về vị trí đứng ở cuối dãy

§ r = n;

§ Hoán vị (a1,ar)

o Bước 2: Loại bỏ phần tử lớn nhất ra khỏi Heap r=r-1;

§ Hiệu chỉnh phần còn lại của dãy từ al đến ar thành một Heap.

o Bước 3:Nếu r >1 ( heap còn phần tử) : lặp lại bước 2

§ Ngược lại: dừng

* *Trường hợp tốt nhất : dãy đã được sắp xếp sẵn=> độ phức tạp là O(1)*
* *Trường hợp xấu nhất : O(nlog2n)*

o Dựa vào tính chất 3, ta có thể thực hiện giai đoạn 1 bằng cách bắt đầu từ heap mặc nhiên an/2+1, an/2+2,…,an, lần lượt thêm vào các phần tử an/2, an/2-1,…,a1. ta sẽ nhận được Heap theo mong muốn. Như vậy giải đoạn 1 tương đương với n/2 lần thực hiện bước 2 của giai đoạn 2.

==>Độ phức tạp về thời gian là O(Logn). Độ phức tạp thời gian của createAndBuildHeap() là O(n) và độ phức tạp thời gian tổng thể của Heap Sort là O(nLogn).