**Ý tưởng**

Heapsort là một thuật toán sắp xếp dựa trên so sánh hoạt động bằng cách xây dựng cấu trúc dữ liệu heap từ mảng đầu vào. Một heap là cây nhị phân trong đó giá trị của mỗi nút lớn hơn hoặc bằng giá trị của các nút con. Thuộc tính này đảm bảo rằng gốc của heap luôn là phần tử lớn nhất trong mảng.

**Mô tả từng bước**

Thuật toán heapsort hoạt động trong hai giai đoạn:

Xây dựng một heap: Giai đoạn này bắt đầu bằng việc tạo ra một heap từ mảng đầu vào. Điều này được thực hiện bằng cách lặp lại so sánh giá trị của mỗi nút với giá trị của các nút con và hoán đổi chúng nếu cần, để duy trì thuộc tính heap.

Sắp xếp mảng: Giai đoạn này bắt đầu bằng việc loại bỏ gốc của heap, là phần tử lớn nhất trong mảng. Gốc sau đó được đặt ở cuối mảng và heap được xây dựng lại mà không có gốc. Quá trình này được lặp lại cho đến khi mảng được sắp xếp.

**Đánh giá độ phức tạp**

Độ phức tạp về thời gian của heapsort là O(n log n) trong trường hợp xấu nhất, trung bình và tốt nhất. Độ phức tạp về không gian của heapsort là O(1), vì nó không yêu cầu bất kỳ không gian bổ sung nào.

**Biến thể / cải tiến**

Có một số biến thể của heapsort có thể cải thiện hiệu suất của nó. Một biến thể là sử dụng việc xây dựng heap từ dưới lên, có thể nhanh hơn so với việc xây dựng heap từ trên xuống được sử dụng trong thuật toán heapsort tiêu chuẩn. Một biến thể khác là sử dụng thuật toán heapsort lai, kết hợp heapsort với thuật toán sắp xếp khác, chẳng hạn như quicksort.

**Kết luận**

Heapsort là một thuật toán sắp xếp nhanh và hiệu quả có độ phức tạp về thời gian trong trường hợp xấu nhất là O(n log n). Nó cũng là một thuật toán sắp xếp ***in place***, có nghĩa là nó không yêu cầu bất kỳ không gian bổ sung nào. Những thuộc tính này khiến heapsort trở thành một lựa chọn tốt để sắp xếp các mảng lớn.

**Nguồn tham khảo**

https://en.wikipedia.org/wiki/Heapsort

https://www.geeksforgeeks.org/heap-sort/

http://syllabus.cs.manchester.ac.uk/ugt/2019/COMP26120/SortingTool/heap\_sort\_info.html