**Câu 1:**Trong các phương pháp xếp thứ tự đã học, phương pháp nào tối ưu nhất, và kém tối ưu nhất? Tại sao?

 Quick Sort là thuật toán sắp xếp nhanh nhất, tối ưu nhất trong phần lớn các trường hợp. Heap Sort là kém tối ưu nhất vì nó là thuật toán chậm nhất và chỉ được dùng trong các trường hợp dữ liệu vào gồm hang triệu phần tử.

**Câu 2:** Trong 2 phương pháp tìm kiếm đã học, trường hợp nào thì cả 02 phương pháp đều như nhau? Giải thích tại sao?

Trong trường hợp tìm kiếm phần tử nhỏ nhất và lớn nhất trong mảng ít phần tử thì cả 2 phương pháp đều như nhau.

**Câu 3:** Ngoài các phương pháp xếp thứ tự đã học, hãy tìm hiểu thêm một phương pháp xếp thứ tự khác, giới thiệu sơ và giải thích.

[Sắp xếp theo cơ số](https://vi.wikipedia.org/wiki/S%E1%BA%AFp_x%E1%BA%BFp_theo_c%C6%A1_s%E1%BB%91) (radix sort) dựa trên tính chất "số" của các khóa. Trong giải thuật sắp xếp theo cơ số, ta không chỉ so sánh giá trị của các khóa, mà so sánh các thành phần của khóa. Giả sử các khóa là các số biểu diễn theo hệ ghi số cơ số M. Khi đó sắp xếp theo cơ số sẽ so sánh từng k

Chúng ta mô tả cách sắp này khi cơ số M=10. Giả sử phải sắp các hồ sơ đánh số bởi 3 chữ số thập phân. Đầu tiên ta chia các hồ sơ vào các đống có cùng chữ số hàng trăm (đồng thới xếp các đống theo thứ tự của chữ số hàng trăm), trong mỗi đống con lại phân chia theo chữ số hàng chục, cuối cùng trong mỗi đống có cùng chữ số hàng trăm và hàng chục, sắp xếp theo thứ tự của chữ số hàng đơn vị.

Trong máy tính, đương nhiên việc sắp xếp theo cơ số [nhị phân](https://vi.wikipedia.org/wiki/H%E1%BB%87_nh%E1%BB%8B_ph%C3%A2n" \o "Hệ nhị phân) (cơ số 2) hoặc cơ số là lũy thừa của 2 là thuận lợi nhất. Trong trường hợp cơ số 2, việc phân hoạch tương tự như phân hoạch trong Quick Sort, chỉ khác ở chỗ cách chọn phần tử chốt.