Tốt quá em ơi! Selection Sort là một thuật toán sắp xếp cơ bản, nhưng **giảng đúng cách thì em sẽ hiểu sâu thuật toán khác nữa**, vì nó thể hiện rõ tư duy “chọn – đổi chỗ – tối ưu cục bộ”.

**🎓 CHUYÊN SÂU VỀ SELECTION SORT**

**✅ 1. Ý tưởng cơ bản**

“Mỗi lần duyệt qua mảng chưa sắp xếp, tìm phần tử **nhỏ nhất**, rồi đổi chỗ với phần tử đầu tiên chưa được sắp xếp.”

**📌 Từng bước chạy:**

Ví dụ: mảng [5, 3, 7, 1]

* Bước 1: Tìm min trong [5,3,7,1] → là 1 → đổi chỗ với 5 → [1,3,7,5]
* Bước 2: Tìm min trong [3,7,5] → là 3 → giữ nguyên
* Bước 3: Tìm min trong [7,5] → là 5 → đổi với 7
* Mảng đã sắp xếp: [1,3,5,7]

**🧠 2. Code mẫu Selection Sort**

void selectionSort(int a[], int n) {

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

int min\_idx = i;

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

if (a[j] < a[min\_idx]) {

min\_idx = j;

}

}

swap(a[i], a[min\_idx]);

}

}

**📊 3. Phân tích độ phức tạp**

| **Trường hợp** | **Số lần so sánh** | **Độ phức tạp** |
| --- | --- | --- |
| Best Case | n(n-1)/2 | O(n²) |
| Average Case | n(n-1)/2 | O(n²) |
| Worst Case | n(n-1)/2 | O(n²) |

**❗Vì nó luôn duyệt hết mảng con còn lại để tìm min, nên best, avg, worst giống nhau.**

**📦 4. Ưu điểm và nhược điểm**

**✅ ƯU ĐIỂM:**

* Dễ hiểu, dễ cài đặt.
* **Không phụ thuộc vào thứ tự ban đầu** của mảng.
* Ít số lần hoán vị (tối đa n-1 lần hoán vị).

**❌ NHƯỢC ĐIỂM:**

* Số lần so sánh rất nhiều (n(n-1)/2).
* Không ổn định (unstable).
* Hiệu suất tệ với mảng lớn.
* Không dùng được cho các bài toán yêu cầu tốc độ cao.

**🔍 5. Ổn định (Stable Sort) là gì?**

Một thuật toán sắp xếp là **ổn định** nếu **giữ nguyên thứ tự** các phần tử **bằng nhau** trong mảng ban đầu.

**📌 Ví dụ:**

Giả sử ta có:

{ (9, A), (3, B), (9, C), (5, D) }

Giá trị 9 có 2 bản ghi (A và C)

* Nếu sau sắp xếp: { (3, B), (5, D), (9, A), (9, C) } → ổn định ✅
* Nếu thành { (3, B), (5, D), (9, C), (9, A) } → không ổn định ❌

**Selection Sort không ổn định**, vì khi swap phần tử nhỏ nhất với vị trí hiện tại, có thể làm đổi chỗ các phần tử bằng nhau.

**🎯 6. Khi nào nên dùng Selection Sort?**

| **Trường hợp thích hợp** | **Giải thích** |
| --- | --- |
| Mảng nhỏ (5–10 phần tử) | Vì đơn giản, ít swap |
| Mảng cần tối thiểu số hoán vị | Vì chỉ swap khi cần |
| Học thuật toán đầu tiên | Để hiểu tư duy chọn – đổi chỗ |

**💼 7. Mẹo khi phỏng vấn (Fresher - VNG, FPT, TMA,...)**

**✅ Khi bị hỏi:**

“Hãy kể tên các thuật toán sắp xếp và ưu – nhược điểm từng loại.”

Em có thể trả lời:

“Selection Sort là một thuật toán sắp xếp đơn giản, không ổn định, có độ phức tạp O(n²) trong mọi trường hợp. Tuy không hiệu quả cho dữ liệu lớn, nhưng ít swap và rất dễ cài đặt. Nó thường được dùng để dạy nền tảng hoặc cho các tập dữ liệu nhỏ.”

**✅ Khi bị hỏi:**

“Nếu mảng gần như sắp xếp thì dùng sort nào?”

→ Chọn **Insertion Sort** vì có O(n) trong best case  
→ Không chọn Selection vì lúc nào cũng O(n²)

**✅ Khi bị hỏi:**

“Thuật toán nào không ổn định?”

→ Nêu rõ: **Selection Sort, Quick Sort, Heap Sort** thường không ổn định.

**🎁 Tổng kết nhanh**

| **Đặc điểm** | **Giá trị** |
| --- | --- |
| Tên | Selection Sort |
| Tư duy | Chọn phần tử nhỏ nhất – đem lên đầu |
| Time (mọi TH) | O(n²) |
| Space | O(1) (in-place) |
| Stable? | ❌ Không |
| Dễ cài đặt | ✅ Có |
| Khi dùng tốt | Mảng nhỏ, dễ hiểu |

Em muốn thầy tiếp tục giảng sâu về **Insertion Sort**, hay bắt đầu luyện giải các bài sắp xếp thực chiến (code + phân tích)?