Câu 1:

### Bảng so sánh 2 prompt và kết quả:

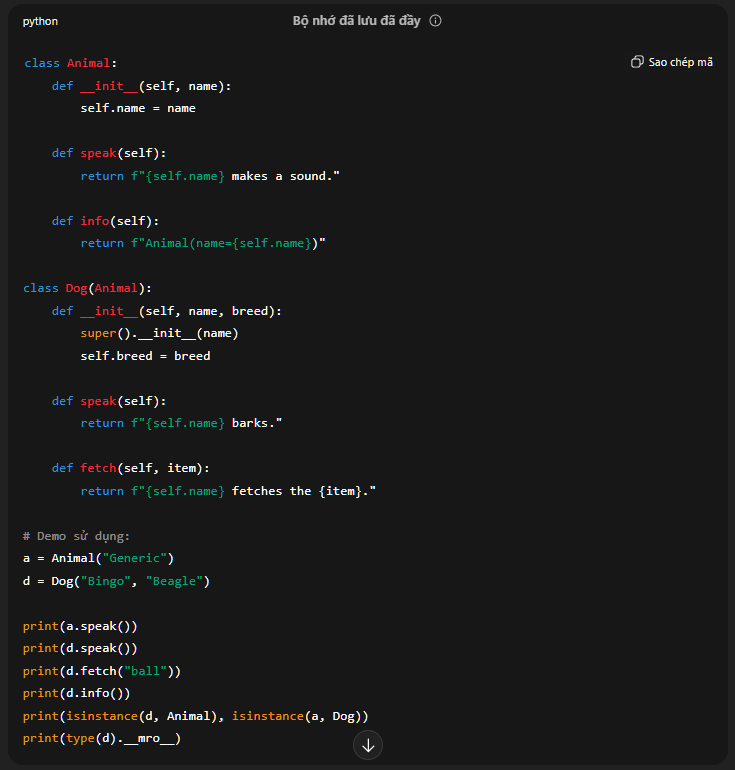
| **Tiêu chí** | **Prompt 1: “Giải thích cấu trúc dữ liệu Stack.”** |  | **Prompt 2: “Tôi chưa biết gì về cấu trúc dữ liệu Stack. Hãy giải thích đơn giản kèm ví dụ đời thực (ví dụ: chồng đĩa).”** |
| --- | --- | --- | --- |
| **Cách diễn đạt** | Giải thích lý thuyết khô khan, mang tính định nghĩa. |  | Giải thích chậm rãi, dễ hiểu, kèm ví dụ gắn với đời sống. |
| **Nội dung trả lời** | Nêu: Stack là cấu trúc dữ liệu dạng LIFO (Last In – First Out), có 2 thao tác chính: push (thêm), pop (lấy ra). |  | Nêu giống trên nhưng diễn đạt nhẹ nhàng hơn, thêm ví dụ **chồng đĩa**: bỏ đĩa mới → đặt lên trên (push), lấy đĩa → lấy cái trên cùng trước (pop). |
| **Mức độ dễ hiểu** | Khó tiếp cận với người mới học, dễ bị ngợp nếu chưa biết thuật ngữ. |  | Thân thiện, dễ hình dung, học tới đâu liên hệ tới thực tế tới đó. |
| **Cảm nhận người học** | “Biết định nghĩa nhưng hơi mơ hồ, chưa rõ hình ảnh.” |  | “À, dễ hiểu rồi, stack giống như chồng đĩa thật, mình nhớ ngay được cách hoạt động.” |

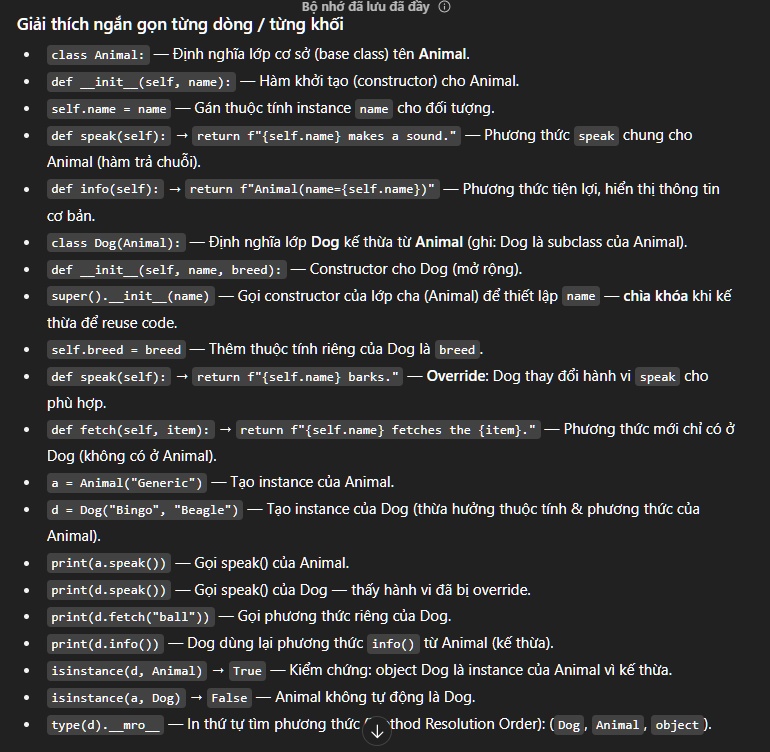
Câu 2:

### Bảng so sánh 2 prompt và kết quả

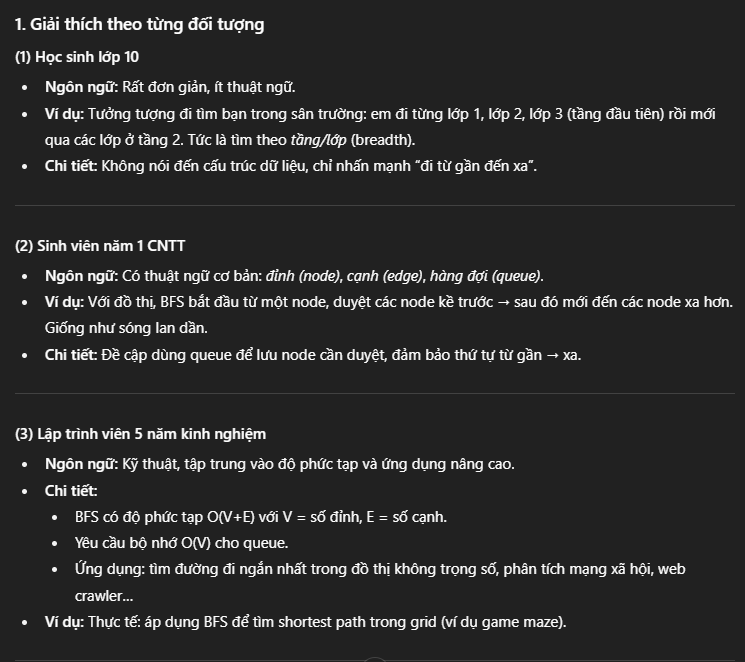
| **Tiêu chí** | **Prompt 1: “So sánh Bubble Sort, Merge Sort và Quick Sort theo tiêu chí: độ phức tạp thời gian, độ phức tạp bộ nhớ, và tình huống áp dụng.”** | **Prompt 2: “… Nếu AI trả lời dạng văn bản → yêu cầu lại: ‘Trình bày kết quả dưới dạng bảng.’”** |
| --- | --- | --- |
| **Cách diễn đạt** | Hỏi trực tiếp, không gợi ý format. | Hỏi thêm, ép AI phải trình bày bằng bảng. |
| **Nội dung trả lời** | AI sẽ viết kiểu liệt kê: Bubble Sort O(n²), Merge Sort O(n log n), Quick Sort trung bình O(n log n)… kèm giải thích. Nhưng đọc hơi dài. | AI gom tất cả thông tin thành **một bảng rõ ràng**: cột = thuật toán, dòng = thời gian, bộ nhớ, tình huống áp dụng → nhìn phát hiểu ngay. |
| **Mức độ dễ hiểu** | Người mới đọc có thể thấy hơi rối vì nhiều chữ, phải tự tóm lại. | Rõ ràng, trực quan, dễ so sánh vì nhìn theo hàng/cột. |
| **Cảm nhận người học** | “Có kiến thức rồi nhưng phải tự lọc.” | “Đỡ mệt, nhìn bảng là nắm ngay điểm khác biệt.” |

Câu 3:





Câu 4:



**So sánh 3 cách giải thích:**

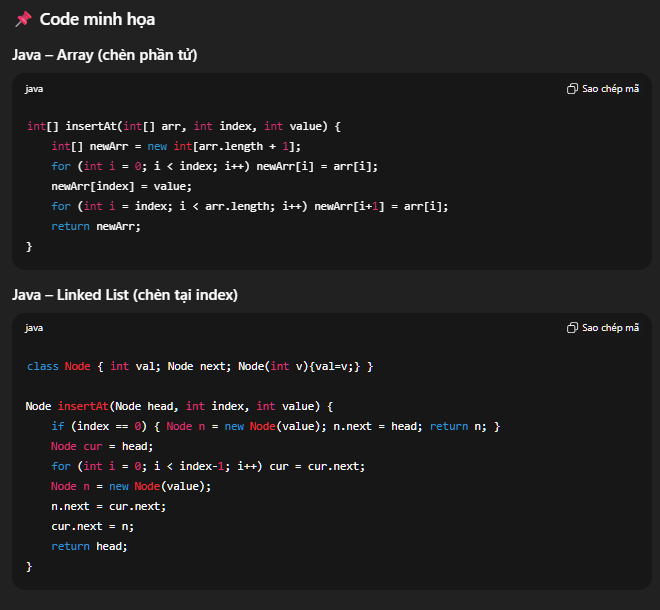
| **Tiêu chí** | **Học sinh lớp 10** | **Sinh viên năm 1** | **Lập trình viên 5 năm** |
| --- | --- | --- | --- |
| **Ngôn ngữ** | Đời thường, không technical | Có technical cơ bản | Kỹ thuật, chi tiết cao |
| **Ví dụ** | Sân trường / lớp học | Đồ thị đơn giản / lan sóng | Maze, mạng xã hội, web crawler |
| **Chi tiết** | Ý tưởng chung | Ý tưởng + công cụ (queue) | Ý tưởng + công cụ + phân tích phức tạp + ứng dụng thực tế |
| **Độ khó** | Rất dễ hiểu | Trung bình | Cao, yêu cầu nền tảng vững |

### Cách phù hợp với mình: Với mình, cách số **(2) – sinh viên năm 1** là phù hợp nhất. Vì nó vừa đủ technical (có nhắc đến queue, node, edge) để hiểu thuật toán rõ ràng, nhưng vẫn dễ hình dung qua ví dụ sóng lan. Nếu chỉ như lớp 10 thì hơi “mù mờ”, còn cách số (3) thì nặng quá, nhiều ứng dụng nâng cao chưa cần tới khi mới học.

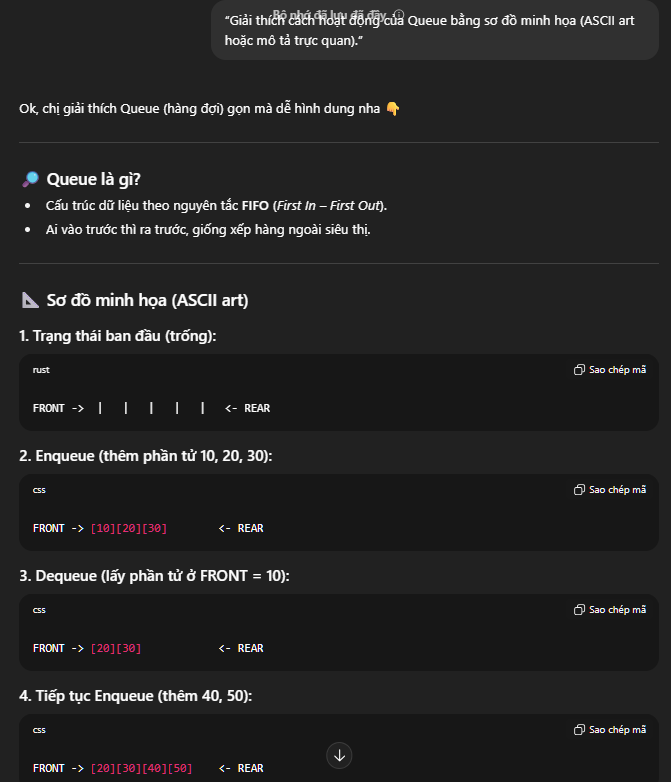
Câu 5:

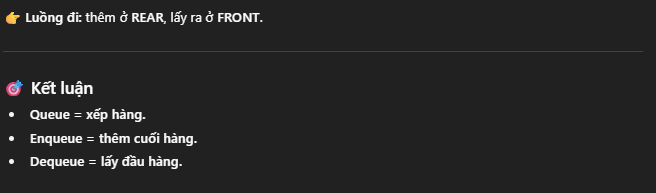
Bảng so sánh:

| **Tiêu chí** | **Array** | **Linked List** |
| --- | --- | --- |
| Truy cập | O(1) theo index | O(n), phải duyệt từng node |
| Chèn/Xóa | O(n) (phải dịch chuyển) | O(1) nếu có con trỏ, O(n) để tìm |
| Bộ nhớ | Liên tục, tiết kiệm | Mỗi node tốn thêm con trỏ |
| Cache | Tốt | Kém |
| Hợp với | Truy cập nhanh, dữ liệu cố định | Nhiều chèn/xóa, kích thước linh hoạt |



Câu 6:





Câu 7:

So sánh:

| **Khía cạnh** | **Prompt 1 (HS cấp 3)** | **Prompt 2 (Junior Dev)** |
| --- | --- | --- |
| **Ngôn ngữ** | Đời thường, ví dụ gần gũi, ít thuật ngữ. | Kỹ thuật, có từ vựng thuật toán + Big-O. |
| **Mức chi tiết** | Cơ bản, tập trung vào “hình dung” hơn là chính xác. | Sâu hơn, có so sánh hiệu năng, giải thích pivot/partition cụ thể. |
| **Công cụ minh họa** | Ví dụ số nhỏ, sơ đồ chia nhóm. | Code Python, ví dụ chạy thật. |
| **Trọng tâm** | Hiểu khái niệm “chia để trị”. | Hiểu lợi ích, áp dụng thực tế vào lập trình. |
| **Mục tiêu học tập** | Làm quen với ý tưởng thuật toán. | Nâng cấp từ thuật toán cơ bản sang thuật toán hiệu quả hơn. |

## AI thay đổi cách giải thích khi bối cảnh khác nhau:

**Context (người học là ai)** quyết định **mức độ trừu tượng**.  
→ Với HS cấp 3: giải thích như kể chuyện.  
→ Với junior dev: giải thích như đồng nghiệp, đưa code.

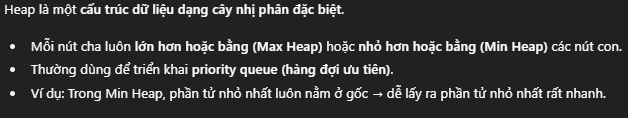
**Điểm tựa kiến thức có sẵn**:  
→ HS cấp 3: chưa có gì, AI xây nền từ đầu.  
→ Junior dev: đã biết Bubble Sort, AI tận dụng sự so sánh để “đòn bẩy” sang Quick Sort.

**Mục tiêu nhận thức**:  
→ HS cấp 3: định hình khái niệm.  
→ Junior dev: ứng dụng + hiệu năng.

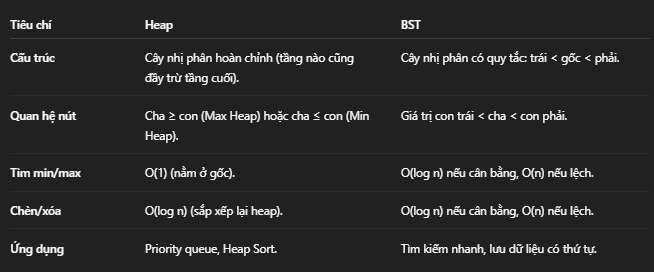
Nói cách khác: **AI điều chỉnh “tone + depth + format” để match background người học.**

Câu 8:

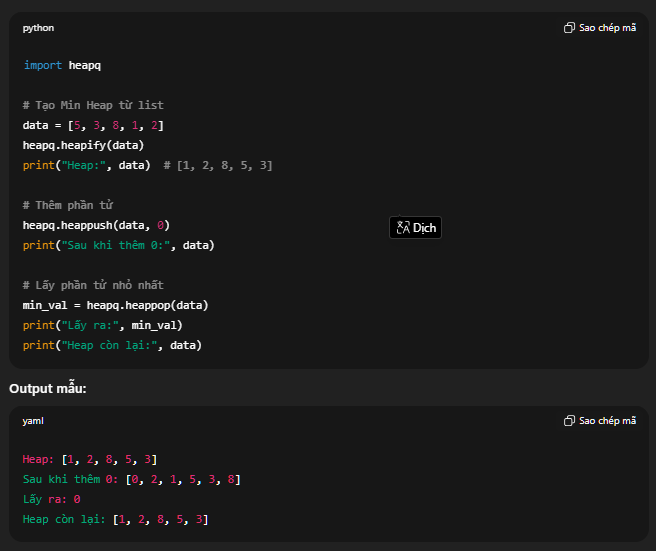
**Prompt 1:**  
“Giải thích Heap là gì theo cách đơn giản, dễ hiểu cho người mới học cấu trúc dữ liệu.”



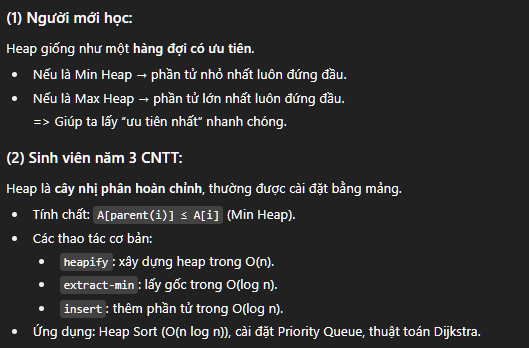
**Prompt 2:**  
“So sánh Heap với Binary Search Tree (BST) bằng bảng.”



**Prompt 3:**  
“Viết ví dụ code Python minh họa cách sử dụng Heap (priority queue).”



**Prompt 4:**  
“Giải thích Heap cho 2 đối tượng: (1) người mới học, (2) sinh viên năm 3 ngành CNTT.”



Câu 9:

