বাংলার বিহার উড়িষ্যার নতুন নবাব, সিংহাসনে আরোহন করিবার পরপরই তাহার পিয়াদাকে ডাকিয়া কহিলেন, "আমার বিশাল ঘোড়ার বহরের জন্য আস্তাবলের(ঘোড়ার রাখিবার স্থান) সন্ধান কর" পিয়াদা কহিল "জো হুকুম জাহাপনা"

নবাবের পিয়াদা নবাবের ঘোড়ার জন্য আস্তাবল সন্ধান শুরু করিলেন, তিনি সন্ধান নিয়া জানিতে পারিলেন, রাজ্যে দুই প্রকারের আস্তাবল বিদ্যমান। আস্তাবলগুলো হইলঃ

<h2>প্রথম প্রকার</h2>

<p>এই প্রকারের আস্তাবলের প্রধান সুবিধা হইল, এইখানে কত সংখ্যক ঘোড়া রাখা যাইবে তাহার কোন সীমা নাই। যত ইচ্ছা তত ঘোড়া রাখা যাইবে। এই কথা শুনিয়া নবাব কহিলেন, "বেশ, বেশ, এত মহা আনন্দের সংবাদ। তা এই আস্তাবলে ঘোড়া রাখিব কি করিয়া?" উত্তরে পিয়াদা কহিল, "এই আস্তাবলের একটি চাবি থাকিবে নবাবের হাতে, নবাব সেই চাবি দিয়া প্রথম ঘোড়ার ঘরে প্রবেশ করিতে পারিবেন। আর প্রথম ঘোড়ার ঘরে হইতে দ্বিতীয় ঘোড়ার ঘরের চাবি, এইভাবে দ্বিতীয় ঘোড়ার ঘরে হইতে তৃতিয় ঘোড়ার ঘরের চাবি। যদি আপনার ঘোড়ার সংখ্যা বাড়িয়া যায় তাহা হইলে শুধুমাত্র নতুন একটি ঘোড়ার ঘর যুক্ত করিতে হইবে আর নতুন ঘরের চাবি সর্বশেষ ঘোড়ার ঘরে রাখিতে হইবে। আবার আপনার যদি ঘোড়ার সংখ্যা কমিয়া যায় তাহা হইলে শুধু মাত্র শেষ ঘোড়ার ঘরটির আগের ঘর হইতে শেষ ঘোড়ার ঘরের চাবিটি ফেলিয়া দিলেই হইবে"। এই পদ্ধতি শুনিয়া রাজা কহিলেন, "তা আমার সাদা ঘোড়াটা যদি আমি সবার শেষ ঘরে রাখি আর এখনই যদি আমি সাদা ঘোড়া নিয়ে বেইর হইতে চাই তাহা হইলে কি করিতে হইবে?" পিয়াদা বলিল, "জাহাপনা, তাহা হইলে আপনাকে সকল ঘোড়ার ঘর হইয়া শেষ ঘরের চাবি নিয়ে তথাপি আপনাকে সাদা ঘোড়া নিয়া বের হইতে হইবে, আরো একটা ব্যাপার জাহাপনা, এই ধরনের ঘোড়ার আস্তাবলে প্রতিটি ঘোরার ঘরে চাবি রাখার জন্য অতিরিক্ত জায়গা প্রয়োজন হইয়া থাকে"। এই কথা শুনিয়া নবাব চিন্তায় পড়িয়া গেলেন। তিনি মনে মনে ভাবিলেন "ইহাতো বড়ই অসুবিধার কথা"। তিনি তথাপি পিয়াদাকে জিজ্ঞেস করিলেন "আর কি প্রকারের আস্তাবল আছে বলো"। পিয়াদা কহিল, "জো হুকুম জাহাপনা"</p>

<img src="https://raw.githubusercontent.com/alimranahmed/resources/master/array\_vs\_linked\_list/linked\_list\_demo.jpg" alt="Linked List demo" style="width:100%">

<h2>দ্বিতীয় প্রকার</h2>

<p>এই প্রকারের আস্তাবলের সকল চাবি থাকিবে নবাবের কাছে। তিনি যখন ইচ্ছা তখন যেই চাবি ইচ্ছা সেই চাবি দিয়া যে কোন ঘোড়ার ঘরে প্রবেশ করিতে পারিবেন। এই কথা শুনিয়াতো নবাব মহাখুশি! তখন পিয়াদা কহিল "জাহাপনা, এই ধরনের আস্তাবলে আপনি কত সংখ্যক ঘোড়া রাখবেন তা আগে থেকে ঠিক করিয়া রাখিতে হইবে। যদি আপনার ঘোড়া সংখ্যা কখনো বাড়িয়া যায় তাহা হইলে কিন্তু নতুন আস্তাবল ক্রয় করিতে হইবে। আর যদি ঘোড়া সংখ্যা কমিয়া যায় তাহা হইলে কিন্তু অকারণে আস্তাবলের জায়গা বিনষ্ট হইবে"। শুনে নবাব কহিলেন "আমার ঘোড়া সংখ্যা গত ১০ বছরে খুব বেশি কমেও নাই বাড়েও নাই, তাই এই প্রকারের আস্তাবলই আমি চাই"। পিয়াদা কহিল, "জো হুকুম জাহাপনা"</p>

<img src="https://raw.githubusercontent.com/alimranahmed/resources/master/array\_vs\_linked\_list/array\_demo.jpg" alt="Array demo" style="width:100%"><br><br>

<p><strong>হা হা হা, উপরের এই নাটকিয়তার সাথে এর‍্যে বা লিংকড লিস্টের কি সম্পর্ক?</strong></p>

<p>সম্পর্ক আছে। এবার আমরা যদি ভাবি, উপরের প্রথম ধরনের আস্তাবল হল, লিংকড লিস্ট আর দ্বিতীয় ধরনের আস্তাবল হলো এর‍্যে, তাহলে ব্যাপারটা কেমন হয়? একেকটা ঘোড়া যদি হয় একেকটা ডাটা ইউনিট তাহলে কেমন হবে? ঠিক তাই, উপরের দুইটা চিত্রই আসলে প্রোগ্রামিং-ই বহুল আলোচিত ডাটা স্টাকচার এর‍্যে এবং লিংকড লিস্টের ধারণার আনুরূপ। </p>

<h2>লিংকড লিস্ট</h2>

<p>এখানে পরিষ্কার ভাবেই বুঝা যাচ্ছে যে, যদি আমি আমাদের কোন ডাটার স্ট্রাকচার হিসাবে লিংকড লিস্টকে বেছে নেই তাহলে, ডাটার পরিমান কখনো কমলে বার বাড়লেও কোন সমস্যা নেই কারণ সে অনুযায়ী লিংকড বড় কিংবা ছোট হয়ে যাবে। কিন্তু আমরা কোন ডাটাকে সরাসরি এক্সেস করতে পারব না। কারণ সব ডাটার চাবি(কী/পয়েন্টার) আমাদের কাছে নেই। এই লিংকড লিস্টের যেহেতু সব ডাটার চাবি আমাদের কাছে নেই সেহেতু এই চাবি নির্ভর এলগরিদম(যেমন, বাইনারী সার্চ) এই ডাটা স্ট্রাকচারে ব্যাবহার করা যাবে না। এই ডাটা স্ট্রাকচারে আমাদের কাছে শুধু মাত্র প্রথম ডাটা ইউনিটের চাবি(Head) থাকে। তাই প্রথম ডাটা ইউনিট ছাড়া অন্য কোন ডাটা ইউনিটকে এক্সেস করতে হলে ঐ প্রথম ডাটা ইউনিটের চাবি দিয়েই একে একে সব ডাটা ইউনিট এক্সেস করতে পারব। কাজেই আমরা বলতে পারি <strong>লিংকড লিস্টের ডাটা এক্সেসের জন্য বেশি সময়ের(Access/Traverse time) প্রয়োজন</strong>। আবার, লিংকড লিস্টে যেহেতু, প্রতেকটি ডাটার ঘরে, পরবর্তি ডাটার লিংক/চাবি রাখতে হয় সেহেতু এই চাবি রাখার জন্য আবার অতিরিক্ত জায়গার প্রয়োজন হয়। তবে এই ডাটা স্ট্রাকচারে যে কোন সময় চাইলেই খুব সহজে নতুন ডাটা যুক্ত(Insert) করা যায় বার কোন ডাটা মুছে(Delete) দিয়ে দেওয়া যায়। কারণ এখানে শুধু একটি ডাটা ইউনিটের ঘর থেকে অন্য ডাটা ইউনিটের চাবি ফেলে দিলেয় যে ডাটা ইউনিটের চাবি ফেলে দিয়েছি সেই ডাটা ইউনিটটি বাদ হয়ে যাবে। তার মানে লিংকড লিস্টে কোন ডাটা যুক্ত করতে কিংবা বাদ দিয়ে খুব কম সময়ের প্রয়োজন হয়।</p>

<h2>এর‍্যে</h2>

<p>এর‍্যেতে কি পরিমান ডাটা রাখার যাবে তার আগে থেকেই নির্ধারণ করা থাকে। ফলে, এতে ঐ নির্ধারিত পরিমানের বেশি ডাটা রাখা যাবে না। পাশাপাশি ঐ নির্ধারিত পরিমানের চেয়ে অনেক কম ডাটা রাখলেও, মেমরী অব্যাবহৃত থাকে বা অপচয় হয়। কিন্তু এর‍্যে ব্যাবহার করলে যে কোন ডাটাকে যে কোন সময় এক্সেস করা যাবে। কাজেই আমরা বলতে পারি <strong>এই ডাটা স্ট্রাকচারের ডাটা এক্সেসের জন্য প্রয়োজনীয় সময়(Access/Traverse time) খুবই কম।</strong> কারণ এর‍্যের সকল ডাটার চাবি থাকে আমাদের হাতে। একই কারণে, চাবি ভিত্তিক এলগরিদম গুলো(যেমন, বাইনারী সার্চ) এর‍্যেতে সহজেই ব্যাবহার করা যায়। কিন্তু এই <strong>এর‍্যের যে কোন জায়গা থেকে কিছু মুছে(Delete) দিতে বা কোন জায়গয়াতে কিছু যুক্ত(Insert) করতে হলে লিংকড লিস্টের তুলনায় বেশি সময় প্রয়োজন হয়।</strong> কারণ এখানে কিছু যোগ করতে বা বাদ দিতে হলে মাঝখানের জায়গা ফাকা হয়ে যাবে বা মাঝখানের জায়গা খালি করতে হবে, এই ব্যাপারটা এখানে কিছুটা জটিলতার সৃষ্টি করে। </p>

<h2>কোনটা কখন ব্যাবহার করব?</h2>

<p>আসলে এর‍্যে এবং লিংকড লিস্ট দুটিই খুবই গুরুত্বপুর্ণ ডাটা স্ট্রাকচার কিন্তু ক্ষেত্র বিশেষে একটি অন্যটির চেয়ে বেশি কার্যকর হতে পারে। যেমন, আমাদের কাছে যদি এমন কোন ডাটার থাকে যেখানে মুহুর্তেই আমাদের নতুন নতুন ডাটা যুক্ত(Insert) করতে হয় কিংবা পুরনো ডাটা মুছে(Delete) ফেলতে হয় কিন্তু ডাটা পড়তে(Read) হয় কম, অথবা যদি আমাদের ডাটা কি পরিমান হবে তার কোন নির্দিষ্ট সীমা আমাদের জানা না থাকে সেক্ষেত্রে এর‍্যের চেয়ে লিংকড লিস্টই বেশি কার্যকরী। অন্যদিকে আমরা যদি আগে থেকেই জানি কি পরিমান ডাটা নিয়ে আমাদের কাজ করতে হবে এবং আমাদের ডাটায় নতুন ডাটা যুক্ত করা বা পুরনো ডাটা মুছে দেওয়ার কাজ কম করতে হয় কিন্তু ডাটা পড়ার কাজ বেশি করতে হয় সেক্ষেত্রে এর‍্যে বেশি কার্যকর ভূমিকা রাখতে পারে।</p>