

Data Structure

Assignment 01

Contest Hints

Get Difference

Explanation:

আপনাকে একটি Singly Linked List ইনপুট নিতে হবে। এরপর ঐ Singly Linked List এর সবচেয়ে বড় / ম্যাক্সিমাম নাম্বার এবং সবচেয়ে ছোট / মিনিমাম নাম্বারের ডিফারেন্স টা বের করে সেটি প্রিন্ট করতে হবে।

Hints

আমরা সি প্রোগ্রামিং এর মডিউলে কোনো একটি নাম্বারের **Array** হতে কীভাবে ম্যাক্সিমাম এবং মিনিমাম নাম্বার টি বের করতে হয় তা শিখেছি। এই প্রবলেমটি ঠিক ঐ রকম। এখানে আমরা শুধুমাত্র **Array** এর পরিবর্তে **Singly Linked List** নিয়ে কাজ করবো। এই প্রবলেমটি মূলত একটি **Singly Linked List** এ প্রত্যেকটি **Element** কে এক্সেস করার / **Singly Linked List** এ ইটারেশন করার বিষয়টির লার্নিং নিশ্চিত করার জন্য দেয়া হয়েছে। মডিউলে দেখানো **Linked List** এর প্রতিটি **Element** এক্সেস করার প্রসেস টি জানতে পারলে এই প্রবলেমটি খুব সহজে সমাধান করা যাবে।

Search

Explanation:

আপনাকে কিছু টেস্টকেস দেয়া হবে। প্রতিটি টেস্টকেসের জন্য একটি Linked List এবং একটি ভ্যালু X দেয়া হবে। আপনাকে বলতে হবে X ভ্যালুটি Linked List এর কয় নাম্বার ইন্ডেক্স এ আছে। যদি X ভ্যালুটি Linked List এ না থাকে, তবে সেই ক্ষেত্রে প্রিন্ট করতে হবে -1।

Hints

এই প্রবলেমটি একটি বেসিক Searching Problem. বিগত কোর্সে আমরা একটি Array তে একটি নির্দিষ্ট ভ্যালু আছে কিনা তা একটি বেসিক লুপ চালিয়ে কীভাবে খুঁজতে হয় তা দেখেছি। এটি ঠিক একই প্রবলেম। এই ক্ষেত্রে আমরা মডিউলে দেখানো উপায়ে Linked List এর প্রতিটি Node কে এক্সেস করে তার ভ্যালুর সাথে আমাদের টার্গেট ভ্যালু X চেক করে দেখবো। পাশাপাশি যেহেতু আমাদের Index প্রিন্ট করতে হবে সেজন্য আমরা একটি ভ্যারিয়েবলের মাধ্যমে index ট্রেক রাখতে পারি।

Same to Same

Explanation:

এই প্রবলেমে আপনাকে দুটি Singly Linked List দেয়া হবে। আপনাকে বলতে হবে, এই দুটি Singly Linked List একদম পুরোপুরি একই কিনা। একই হলে Yes প্রিন্ট করবেন, অন্যথায় No প্রিন্ট করতে হবে।

Hints

Linked List এর প্রতিটি প্রবলেম আমরা **Array** এর সাথে তুলনা করে সমাধান করার চেষ্টা করতে পারি। প্রথমে আমরা দুটি লিঙ্কড লিস্ট ইনপুট নিবো। দুটি **List** আলাদা ভাবে ট্রেক রাখার জন্য আমরা ভিন্ন **Head** পয়েন্টার রাখতে পারি। এরপর দুটি **List** একই কিনা চেক করার জন্য আমরা একটি বেসিক লুপ ব্যবহার করে ঐ দুটি লিঙ্কড লিস্ট এর প্রতিটি **Node** এক্সেস করে তাদের ভ্যালু কম্পেয়ার করতে পারি। মনে করুন, একটি সিচুয়েশন আসলো এমন যে, দুটি **Linked list** কম্পেয়ার করার সময়, প্রথম **List** এর আগে দ্বিতীয় **List** এর **Node** শেষ, এই ক্ষেত্রে আমরা দ্বিতীয় **Linked list** এর ভ্যালু এক্সেস করতে গেলে **Segmentation fault Error** এর সম্মুখীন হবো। তাই এই বিষয়ে আমাদের সতর্ক থাকতে হবে। এইক্ষেত্রে এই কর্নার কেসটি সমাধানের জন্য আমরা আগে থেকে সাইজ বের করে চেক করতে পারি, বা **Node** এ **Iteration** করার ক্ষেত্রে শর্ত ব্যবহার করতে পারি।

Queries

Explanation:

মনে করুন, আপনার কাছে একটি Singly Linked List এ আছে যা শুরুতে খালি অবস্থায় আছে। এরপর আপনাকে কিছু কুয়েরি দেয়া হলো।

কুয়েরি সমূহ:

অপারেশন নং ১: এই ক্ষেত্রে একটি ভ্যালু দেয়া হবে , ঐ ভ্যালুটি Linked List টির Head পজিশন Insert করতে হবে ।

অপারেশন নং ২: এই ক্ষেত্রে একটি ভ্যালু দেয়া হবে , ঐ ভ্যালুটি Linked List টির Tail পজিশনে Insert করতে হবে

অপারেশন নং ৩: এই ক্ষেত্রে একটি Index দেয়া থাকবে , ঐ Index এর Node টি ডিলিট করে দিতে হবে। এইক্ষেত্রে Invalid Index দেয়া থাকলে কোনো অপারেশন করতে হবে না। শুধু মাত্র Linked List টি প্রিন্ট করে দিলেই হবে।

প্রতিটি অপারেশন শেষে ঐ Updated Linked List টি প্রিন্ট করতে হবে ।

অপারেশন শেষে Linked List খালি থাকলে কিছু প্রিন্ট না করে Newline প্রিন্ট করতে হবে।

Hints

Singly Linked List এর মডিউল গুলোতে আমরা কীভাবে লিস্ট এর **Head** এবং **Tail** এ **Node Insert** করতে হয় , কীভাবে কোনো একটি পজিশনের ভ্যালু **Delete** করতে হয় তা জেনেছি। উক্ত প্রবলেমটি উল্লেখ্য অপারেশন গুলো ইমপ্লিমেন্ট এবং এগ্লাই করার মাধ্যমে খুব সহজে সমাধান করে ফেলা যাবে। কিছু কিছু **Exception** যেমন **Head** এ ভ্যালু **Insert** বা **Head Node Delete** করা , **Tail** ট্রেক রাখার মাধ্যমে **Tail** এ কম টাইম কমপ্লেক্সিটিতে ভ্যালু **Insert** করা এই বিষয়গুলো খেয়াল রাখতে হবে। পাশপাশি এমন কোনো **Node** এক্সেস করছি কিনা , যা আসলে **available** নেই , ঐ বিষয় গুলো খেয়াল রাখতে হবে।

Remove Duplicate

Explanation:

আপনাকে একটি Singly Linked List দেয়া হবে। ঐ Singly Linked List এ যেসব নান্বার গুলা Duplicate অর্থাৎ একের অধিক আছে তা Delete করে দিতে হবে। এবং সবশেষে ঐ Linked List টি প্রিন্ট করে দেখাতে হবে।

Hints

Constraints এ N এর মান দেয়া আছে ১০০০। অর্থাৎ এই সমস্যাটি সমাধান করার জন্য আমরা $O(N^2)$ এর **Code** লিখতে পারবো।

লিংকড লিস্ট টির প্রতিটি **Node** এর জন্য, তার পরবর্তী **Node** গুলাতে এই ভ্যালু গুলা চেক করে দেখবো, যদি দেখি পরবর্তী **Index** গুলাতে কারেন্ট ভ্যালুটি ডুপ্লিকেট আছে তবে মডিউলে দেখানো **Node Delete** এর ফাংশনের সাহায্যে ঐ **Node** টি ডিলিট করে দিবো। ডিলিট করার সময় খেয়াল রাখতে হবে নোডটা কি শুরুর নোড নাকি মাঝের নাকি শেষের নোড এবং নোড এর অবস্থান অনুযায়ী ডিলিট অপারেশনটা করতে হবে। যেহেতু লিংকড লিস্ট সর্টেড নাই তাহলে কাজটা এইভাবে করতে হবে। এবং সবশেষে **Linked List** টি প্রিন্ট করে দিবো।