

প্রোগ্রামার এর মত চিন্তা করবেন ? প্রবলেম সলভিং ?



Sharif Hasan

• August 23, 2019

সর্বশেষ আপডেট January 1, 2021

0

1,291

পড়তে 4 মিনিট লাগতে পারে



আপনি যদি একজন প্রোগ্রামার হন তবে আপনি সম্ভবত একটি উক্তি শুনে থাকবেন –

“Everyone in this country should learn to program a computer, because it teaches you to think.” — Steve Jobs

বাংলায় – এই দেশের সকলকেই প্রোগ্রামিং শিখা উচিত, কারন এটা চিন্তা করতে শিখায়।



Problem Solving

আপনি হয়ত ভাবতে পারেন মানে কী ? প্রোগ্রামারের মত চিন্তা ? কেন বা কিভাবে আমি করব ? মূলত সমস্যা সমাধান বা **problem solving** এর জন্য এটা করা দরকার । এই পোস্টে আমার উদ্দেশ্য হলো এটা কিভাবে করতে হবে তা আপনাদের দেখাবো । মজা huh ?

সবার শেষে আপনি জানবেন কি কি পদক্ষেপ নিলে ভালো প্রবলেম সলভার হওয়া যায়

এটা কেন জরুরি?

প্রবলেম সলভ করার দক্ষতা হলো একটা মেটা-দক্ষতা (Meta-Skill) .

আমাদের সকলেরই সমস্যা আছে । আমরা কিভাবে সমাধান করি ? আমাদের মাথা কিভাবে এত সহজে সমস্যার সমাধান করে ফেলে । উত্তর??

আপনার যদি কোন সিস্টেম না থাকে তবে আপনি কিভাবে সমস্যার সমাধান কনতেন ? এখানে কিছু উদাহরন যা প্রথমে আমি করতাম:

- একটা সলুশন চেস্টা করতাম
- ঐটা না হলে আরেকটা চেস্টা করতাম
- তবুও না হলে উপরের টা আবার রিপিট করতাম

হ্যা , উপরের উপায় গুলোতে কাজ হতে পারে । কিন্তু বিশ্বাস করুন প্রবলেম সলভ করার জন্য এটা সবচেয়ে খারাপ উপায় । প্রচুর সময় নষ্ট ।

ভাল রাস্তা হলো একটা ফ্রেমওয়ার্ক থাকা এবং সে অনুযায়ী কাজ করা ।

- প্রায় সকল নিয়োগ দাতাই প্রবলেম সলভিং দক্ষতাকে অগ্রাধিকার দেয় ।
- যতই প্রোগ্রামিং ল্যাংগুয়েজ এর উপর দখল থাকুক না কেন , সবকিছুর উর্ধে প্রবলেম সলভিং এর দক্ষতাকে গুরুত্ব দেয়া হয় ।
- অনেক বড় বড় সমস্যা কে ভেগে ছোট ছোট করে সহজে সমাধান করার দক্ষতা চাকরি পাওয়ার জন্য বেস দক্ষতার চেয়েও বেশি মূল্যবান ।

একটা ফ্রেমওয়ার্ক থাকা

একটা ফ্রেমওয়ার্ক আপনাকে সহায়্য করবে বড় সমস্যাকে সহজেই সমাধান করতে

“The biggest mistake I see new programmers make is focusing on learning syntax instead of learning how to solve problems.” — V. Anton Spraul

বাংলা

– নতুন প্রোগ্রামারদের মধ্যে যে সমস্যা আমি সবচেয়ে বেশি দেখি তা হল

প্রবলেম সলভিং চেয়ে প্রোগ্রামিং সিনটেক্স এর উপরে বেশি গুরুত্ব দিয়ে ফেলা।

এখন, আপনি কি করবেন যখন আপনি কোন সমস্যার মুখোমুখি হবেন?
নিচের পয়েন্ট গুলো গুরুত্বপূর্ণ:

- বুঝতে পারা : বুঝতে হবে সমস্যায় কি বলা হয়েছে ।
জটিল সমস্যা গুলো জটিল হয় এই কারনে যে এখানে বুঝাই যায় না কি জিজ্ঞাসা করা
হয়েছে ? huh ? right ? এই জন্যই এটা প্রথম ধাপ । কখনও এমনও হয় যে একটা সমস্যায় আটকাই গেছি । কোন ভাবেই পারতেছি না । যখন নিজে নিজেই ব্যাখ্যা করি
আর একটা সুক্ষ চোরা ছিদ্র দেখি লজিকে তখন বলেন , কেমন ভা লগে ?? মনডায় কয়
?? এই অনুভূতি টা প্রায় সব প্রোগ্রামারের ই জানা ।
- পরিকল্পনা:
যদি সমস্যা বুঝে ফেলেন তবে কোন পরিকল্পনা ছাড়াই দয়া করে সমস্যা সমাধানে ঝাপ দিবেন না । অর্ধেক রাস্তায় সব গুলিয়ে খেয়ে ফেলা ছাড়া উপায় থাকবে না ।
যদি আপনি উপযুক্ত পথ গুলো কোড করতে না পারেন তবে আপনার গতি ঐ শেষ লাইন
পর্যন্তই আটকে যাবে । এটা প্রোগ্রামিং , এখানে মাথাকে সময় দিতে হবে কিভাবে প্রোগ্রাম এগোবে এবং কিভাবে ডাটা প্রসেস হবে ।
- ভাগ করো শাসন করো:
ব্রিটিশ দের এই উপায়টা মনে হয় সবারই জনা । এই পথে ব্রিটিশরা ভারতবর্ষকে ২০০ বছর শাসন করেছে । যাক গে , কাজের কথায় আসি । প্রবলেম সলভিং এ এটা
সবচেয়ে গুরুত্বপূর্ণ । যদি সম্পূর্ণ সমস্যা একবারে সমাধানে নেমে পড়েন তবে আমি বলব কিছু করার নেই । কাঁদতে হবে । সমস্যাগুলোকে কিছু ছোটো ছোটো খন্ডে
ভাগ করে নিয়ে বড় সমস্যা গুলোকে সহজেই সমাধান করা যায় । ছোট ছোট বলতে এমন
ভাবে ভাগ করতে হবে যাতে একটা ভাগ আরেকটার উপর সরাসরি নির্ভর না করে । যখন
সবগুলো সমস্যার সমাধান হয়ে যাবে তখন সবগুলো কে জোড়া লাগান ।

"If I could teach every beginning programmer one problem-solving skill, it would be the 'reduce the problem technique.'

For example, suppose you're a new programmer and you're asked to write a program that reads ten numbers and figures out which number is the third highest. For a brand-new programmer, that can be a

*tough assignment,
even though it only requires basic programming syntax.*

*If
you're stuck, you should reduce the problem to
something simpler.
Instead of the third-highest number, what about finding
the highest
overall? Still too tough? What about finding the largest
of just three
numbers? Or the larger of two?*

*Reduce the
problem to the point where you know how to solve it
and write the
solution. Then expand the problem slightly and rewrite
the solution to
match, and keep going until you are back where you
started.” — [V. Anton Spraul](#)*

শুধু শুধু দিলাম একটা!

- আটকে গেছেন: আপনি হয়তো এখন ভাবছেন , ভালো হে হে । ধুর ব্যাটা এখন যদি আমি ভাগকরা সমস্যা গুলোতেও আটকে যাই । huh?
- 1. সমস্যা
নাই | এই সমস্যা প্রথমে সবারই হয় । জোড়ে একটা শ্বাস নিয়ে কাজে লেগে যান ।
গ্রেট প্রোগ্রামারদের থেকে আমার পার্থক্য গুলোর একটা হলো আমি ভুল দেখে
বিরক্ত বোধ করলেও তারা করে না । তারা ডিবাগিং করতে আরও ভালবাসে ।
- 2. Debug:
প্রোগ্রামাররা এটাকে ডিবাগ বলে । আপনার প্রোগ্রামের লাইন বাই লাইন ধরে
আগান । দেখেন কোন লাইনে ভুল হয়েছে । এই কাজটা ডিবাগার এর মাধ্যমে অনেকখানি
সহজ হয়ে যায় । “The art of debugging is figuring out what you really told
your program to do rather than what you thought you told it to do.” —
Andrew Singer
- 3. আবার যাচাই করা: একথা পিছিয়ে যান ।
প্রবলেমটা আবার অন্য দৃষ্টিকোন থেকে দেখুন আপনার লজিক ঠিক আছে কিনা? আপনার
লজিক কে ভাঙা যেতে পারে এমন কি কিছু আছে ? **“Sometimes we get so
lost in the details of a problem that we overlook general principles
that would solve the problem at a more general level. [...] The
classic example of this, of course, is the summation of a long list of
consecutive integers, $1 + 2 + 3 + \dots + n$, which a very young Gauss
quickly recognized was simply $n(n+1)/2$, thus avoiding the effort of
having to do the addition.” — C. Jordan Ball**

চেস্টার ওপর চেস্টা করা

একসপ্তাহ প্রবলেম সলভ করে একেবারে টপলিস্টে থাকার চিন্তা করাও পাপ্!

আপনি যদি সত্যি ভালো প্রবলেম সলভার হতে চান তবে প্রচুর সমস্যা সমাধান করুন।
সমাধান সমাধান আর সমাধান। করতে থাকলে দেখবেন একসময় মাথা অটোমেটিক
বলতে

থাকবে আমি গানটলেটের তুড়ি মেরে সব সমস্যা কে ভূষি করে দিব !! তখন টপকোডার
বা রেডকোডার হওয়ার জন্য মাঝখানে থাকবে শুধুই সময়।

তথ্যসূত্র : freecodecamp.org, [Wikipedia](https://wikipedia.org), [Book:How to think like a programmer](#)

ধন্যবাদ : [Abrar Nafee Akhand](#)

লেখাটি কেমন লেগেছে আপনার?

রেটিং দিতে হার্টের উপর ক্লিক করুন।



গড় রেটিং 4.2 / 5. মোট ভোট: 46

#কম্পিউটার টেকনোলজি

#প্রবলেম সলভিং

#প্রোগ্রামিং ভাষা গাইডলাইন