

পাইথনে
ডেটা স্ট্রাকচার ও
অ্যালগরিদম

পাইথনে
ডেটা স্ট্রাকচার ও
অ্যালগরিদম
জাকির হোসাইন

❀ ডায়ালিপি

উৎসর্গ

আবদুল মান্নান স্যারকে

সূচি

ভূমিকা	১১
সংক্ষেপে পাইথন পরিচিতি	১৩
পাইথন ইন্সটল এবং ব্যবহার	১৩
কমেন্ট	১৩
ইন্ডেন্টেশন	১৪
ভ্যারিয়েবল	১৫
কাস্টিং	১৬
ডেটা ইনপুট	১৬
অ্যারিথমিটিক অপারেটর	১৭
কম্পারিজন অপারেটর	১৮
লজিক্যাল অপারেটর	১৯
If স্টেটমেন্ট	২১
while লুপ	২১
for লুপ	২২
লিস্ট	২২
ফাংশন	২৪
পাইথন স্ট্যান্ডার্ড লাইব্রেরি	২৫
অ্যালগরিদম, কমপ্লেক্সিটি অ্যানালাইসিস ও নোটেশন	২৬
অ্যালগরিদম কি	২৬
সুডোকোড	২৮

কমপ্লেক্সিটি অ্যানালাইসিস	২৮
ইন্ট্রাকশন কাউন্টিং	২৯
Best-case analysis	৩০
Worst-case analysis	৩০
Asymptotic Notation	৩১
Tight Bound	৩২
Big O Notation	৩২
বিভিন্ন ধরনের কমপ্লেক্সিটি	৩২
ডেটা স্ট্রাকচার পরিচিতি	৩৫
ডেটা স্ট্রাকচার	৩৫
লিনিয়ার ডেটা স্ট্রাকচার	৩৫
নন-লিনিয়ার ডেটা স্ট্রাকচার	৩৭
ডেটা স্ট্রাকচারের অপারেশনগুলো	৩৮
স্ট্যাক	৪০
স্ট্যাক কী	৪০
পাইথন ইমপ্লিমেন্টেশন	৪১
কিউ	৪৫
কিউ	৪৫
কিউয়ের অপারেশন	৪৫
পাইথনে ইমপ্লিমেন্টেশন	৪৬
লিঙ্কড লিস্ট	৪৯
লিঙ্কড লিস্ট ইমপ্লিমেন্টেশন	৫০
সিঙ্গেল লিঙ্কড লিস্ট	৫৫
ডাবল লিঙ্কড লিস্ট	৫৫
সার্কুলার লিঙ্কড লিস্ট	৫৭

গ্রাফ	৫৮
গ্রাফ	৫৮
গ্রাফের প্রকারভেদ	৫৯
গ্রাফ রিপ্রেজেন্টেশন	৬১
গ্রাফ ট্রান্সার্সাল	৬৭
ব্রেডথ ফাস্ট সার্চ	৬৭
ডেপথ ফাস্ট সার্চ	৭২
ট্রি	৭৮
ট্রি ডেটা স্ট্রাকচার	৭৮
ট্রি-এর টারমিনলজি	৭৯
বাইনারি ট্রি	৮৬
ট্রি ট্রান্সার্সাল	৮৮
প্রি অর্ডার ট্রি ট্রান্সার্সাল	৮৮
ইন অর্ডার ট্রি ট্রান্সার্সাল	৮৯
পোস্ট অর্ডার ট্রি ট্রান্সার্সাল	৯১
গ্রাফ এবং ট্রি-এর মধ্যে পার্থক্য	৯৪
লিনিয়ার সার্চ	৯৫
লিনিয়ার সার্চের টাইম কমপ্লেক্সিটি	৯৬
পাইথনে লিনিয়ার সার্চ	৯৭
বাইনারি সার্চ	৯৯
পাইথনে বাইনারি সার্চ	১০২
বাইনারি সার্চের টাইম কমপ্লেক্সিটি	১০৩
বাবল সর্ট	১০৪
বাবল সর্টের টাইম কমপ্লেক্সিটি	১০৬
পাইথনে বাবল সর্ট	১০৬

সিলেকশন সর্ট	১০৮
পাইথনে সিলেকশন সর্ট	১১২
সিলেকশন সর্টের টাইম কমপ্লেক্সিটি	১১২
রিকারশন	১১৪
রিকার্সিভ অ্যালগরিদম	১১৪
ডিভাইড ও কনকার স্ট্রাটেজি	১১৯
ডিভাইড এবং কনকার স্ট্রাটেজি	১১৯
মার্জ সর্ট	১২০
পাইথনে মার্জ সর্ট	১২৪
কুইক সর্ট	১২৭
পাইথনে কুইক সর্ট	১৩১
গ্রিডি অ্যালগরিদম	১৩৪
কয়েন সমস্যা	১৩৪
গ্রিডি এপ্রোচের সুবিধা	১৩৭
গ্রিডি এপ্রোচের অসুবিধে	১৩৭
ডাইনামিক প্রোগ্রামিং	১৩৮
ডাইনামিক প্রোগ্রামিংয়ের বটম আপ পদ্ধতি	১৪২
রিকার্সন ও ডাইনামিক প্রোগ্রামিং	১৪৩

ভূমিকা

কম্পিউটারের হচ্ছে এমন একটি মানব সৃষ্ট আবিষ্কার যার কাজ সংগৃহীত তথ্যকে প্রক্রিয়াকরণ করে তা আউটপুট হিসেবে প্রদর্শন করা। কম্পিউটার একটি যুগান্তকারী আবিষ্কার হলেও এর নেই কোনো নিজস্ব জ্ঞান। ফলাফল স্বরূপ কম্পিউটারকে সঠিকভাবে পরিচালনা করার জন্য অর্থাৎ তথ্যকে আশানুরূপভাবে প্রক্রিয়াকরণের জন্য পূর্বনির্ধারিত নির্দেশনার প্রয়োজন হয়। পূর্বনির্ধারিত নির্দেশাবলি অনুসরণ করে সংগৃহীত তথ্যের উপর গাণিতিক এবং লজিক্যাল অপারেশন সম্পাদনের মাধ্যমে কম্পিউটার যেকোনো সমস্যা খুব কম সময়ে এবং নির্ভুলভাবে সমাধান করতে সক্ষম। কোনো একটি সমস্যা সমাধানের জন্য পূর্বনির্ধারিত নির্দেশাবলিকেই কম্পিউটার জগতের ভাষায় অ্যালগরিদম বলে। একটি সমস্যার সমাধান অনেকগুলো হতে পারে। অর্থাৎ যেকোনো সমস্যা সমাধানের জন্য ভিন্ন ভিন্ন অ্যালগরিদম তৈরি করা সম্ভব। কিন্তু কোন অ্যালগরিদম খুব কম সময় এবং কম মেমরি স্পেইস ব্যবহার করে সমস্যা সমাধান করতে সক্ষম তা নির্ণয় করা অত্যন্ত গুরুত্বপূর্ণ। কারণ কম্পিউটারে অসীম মেমরি স্পেইস থাকে না। একই সাথে কোনো সমাধান সম্পাদনের জন্য কম্পিউটার যদি অতিরিক্ত সময় ব্যয় করে তাহলে তা কখনো উপযুক্ত সমাধান হিসেবে বিবেচনা করা যায় না। অন্যদিকে অ্যালগরিদম অনুসারে যেহেতু তথ্য প্রক্রিয়াকরণ করা হয় সেহেতু তথ্যকে সুশৃঙ্খল ভাবে সাজিয়ে রাখাও খুবই গুরুত্বপূর্ণ। অন্যথায় কম্পিউটার অ্যালগরিদম অনুযায়ী সঠিক সময় সঠিক তথ্য প্রক্রিয়াকরণ করতে ব্যর্থ হবে। তথ্যকে সুশৃঙ্খলভাবে সাজিয়ে রাখার পদ্ধতিসমূহ ডেটা স্ট্রাকচারে মূল আলোচনা বিষয়।

প্রোগ্রামিংয়ের পাশাপাশি ডেটা স্ট্রাকচার এবং অ্যালগরিদমে স্কিল অর্জন করা অনেকটা অবশ্যক। প্রোগ্রামিং সমস্যার সমাধান শিখায় অন্যদিকে

ডেটা স্ট্রাকচার এবং অ্যালগরিদম, সমস্যার উপযুক্ত সমাধান নির্ণয়ের সহায়তা করে। ছোট বা বড় যেকোনো সফটওয়্যার কোম্পানির ইন্টারভিউতে ডেটা স্ট্রাকচার এবং অ্যালগরিদমের জ্ঞান, প্রোগ্রামিংয়ের স্কিলের তুলনায় বেশি প্রাধান্য দিয়ে থাকে। আমাদের দৈনন্দিন জীবনের কথাই চিন্তা করি, আমরা সে সকল ব্যক্তিকেই বেশি প্রাধান্য দিয়ে থাকি যারা কম সময়ে, কম রিসোর্স ব্যবহার করে সেবা প্রদান করতে সক্ষম। ঠিক তেমনি সফটওয়্যার কোম্পানিগুলো আবশ্যিক এমন কাউকে নিয়োগ করতে চাইবে যে কি না যেকোনো সমস্যার, এমন একটি সমাধান খুঁজে নিতে পারবে যেটির সম্পাদন সময় খুবই কম এবং একই সাথে স্বল্প রিসোর্স ব্যবহার করে। ডেটা স্ট্রাকচার এবং অ্যালগরিদমের স্কিল অর্জনের জন্য কোনো নির্দিষ্ট প্রোগ্রামিং ল্যাঙ্গুয়েজের প্রয়োজন হয় না। যেকোনো প্রোগ্রামিং ল্যাঙ্গুয়েজে ব্যবহার করে ডেটা স্ট্রাকচার এবং অ্যালগরিদম স্কিল ডেভেলপ করা সম্ভব। উক্ত বইটিতে পাইথন ল্যাঙ্গুয়েজ ব্যবহার করে ডেটা স্ট্রাকচার এবং অ্যালগরিদমের যাবতীয় সকল খুঁটিনাটি উদাহরণসহ বর্ণনা করা হয়েছে। তবে যে কেউ চাইলে অন্যান্য প্রোগ্রামিং ল্যাঙ্গুয়েজ ব্যবহার করে খুব সহজেই উক্ত বইয়ের উদাহরণগুলো ব্যবহার করে ডেটা স্ট্রাকচার এবং অ্যালগরিদম সম্পর্কে পর্যাপ্ত জ্ঞান অর্জন করতে পারবে।

মোঃ শরীফ উদ্দিন

প্রভাষক

কম্পিউটার সায়েন্স অ্যান্ড ইঞ্জিনিয়ারিং

ইউনিভার্সিটি অব ক্রিয়েটিভ টেকনোলজি চট্টগ্রাম।

“Big things have small beginnings”

~ T.E. Lawrence

অধ্যায় ১

সংক্ষেপে পাইথন পরিচিতি

পাইথন ইন্সটল এবং ব্যবহার

পাইথন প্রোগ্রাম কম্পিউটারে রান করার জন্য পাইথন ইন্সটল করে নিতে হয়। ইন্সটলার পাওয়া যাবে python.org/downloads ঠিকানায়। ইন্সটল করে নেওয়ার পর যেকোনো কোড এডিটর অথবা আইডিই ব্যবহার করে পাইথন প্রোগ্রাম লেখা এবং রান করা যাবে। পাইথন প্রোগ্রাম লেখা এবং রান করার জন্য জনপ্রিয় পাইথন আইডিই হচ্ছে PyCharm। পাইচার্ম পাওয়া যাবে jetbrains.com/pycharm ঠিকানায়।

এছাড়া পাইথন কম্পিউটারে ইন্সটল না করেও অনলাইনে পাইথন প্রোগ্রাম লেখা এবং রান করা যাবে। অনেকগুলো অপশনই রয়েছে। আমি সাজেস্ট করব গুগলের কোল্যাব। colab.research.google.com ঠিকানায় গিয়ে একটা জুপিটার নোট বুক তৈরি করে পাইথন প্রোগ্রাম রান লেখা এবং রান করা যাবে।

কমেন্ট

কোন কোড কী কারণে লেখা হয়েছে, তা লিখে রাখার জন্য প্রোগ্রামিংয়ে কমেন্ট ব্যবহার করা হয়।

পাইথনে এক লাইনের কমেন্ট লেখার জন্য হ্যাশ (#) ব্যবহার করা হয়। যেমন:

```
# printing hello world!  
print("Hello World!")
```

একের অধিক লাইনের কमेंট লিখতে চাইলে তিনটা ডাবল কোটেশনের মধ্যে কमेंট লিখতে হয়।

যেমন:

```
"""
This is our first Python program.
In this program, we will print Hello World!
in the console.
"""
print("Hello World!")
```

ইন্ডেন্টেশন

ইন্ডেন্টেশন হচ্ছে হোয়াইট স্পেস। পাইথনে কোড লেখার সময় এই ইন্ডেন্টেশনগুলো মেনে কোড লিখতে হয়। কিছু কিছু প্রোগ্রামিং ল্যাঙ্গুয়েজে ইন্ডেন্টেশন না মানলেও প্রোগ্রাম ঠিকমতো কাজ করে। কিন্তু পাইথনে ইন্ডেন্টেশন ঠিকমতো না দেওয়া হলে প্রোগ্রামে এরর দিবে। যে আউটপুট পেতে চাচ্ছি, তেমন আউটপুটও পাওয়া যাবে না।

যেমন নিচের কোডগুলো দেখি:

```
5
x = 21
y = 19
if x > y:
    print "x is greater than y"
```

এখানে print "x is grater then y" স্টেটমেন্টের আগে পাঁচটা স্পেস রয়েছে। আমরা যদি এই স্পেসগুলো না দিয়ে প্রোগ্রামটি লিখে রান করি, তাহলে প্রোগ্রামে এরর দিবে। কিন্তু স্পেসগুলো দেওয়ার পর যদি রান করি, তাহলে সুন্দর মতো প্রোগ্রামটা রান হবে।

ভ্যারিয়েবল

প্রোগ্রামে কোন ডেটা সংরক্ষণ করে রাখার জন্য ভ্যারিয়েবল ব্যবহার করা হয়। যেমন:

```
price = 550
g = 9.8
book_name = 'Data Structure and Algorithms in Python'
is_leap_year = False
```

উপরের উদাহরণে

- price হচ্ছে একটা ইন্টিজার ভ্যারিয়েবল। ইন্টিজার বলতে পূর্ণসংখ্যাকে বুঝায়। যে সংখ্যায় দশমিক থাকবে না।
- g হচ্ছে ফ্লোটিং পয়েন্ট ভ্যারিয়েবল (দশমিক যুক্ত সংখ্যা)
- book_name হচ্ছে স্ট্রিং ভ্যারিয়েবল (a sequence of characters)। স্ট্রিং ডাবল কোট অথবা সিঙ্গেল কোটের ভেতর লিখতে হয়।
- is_leap_year হচ্ছে বুলিয়ান ভ্যারিয়েবল। বুলিয়ানের ভ্যালু True অথবা False হবে।

ভ্যারিয়েবলে মাল্টিপল অ্যাসাইনমেন্ট

এক লাইনে একাধিক ভ্যারিয়েবল কোন ভ্যালু অ্যাসাইন করার জন্য মাল্টিপল অ্যাসাইনমেন্ট ব্যবহার করা হয়।

```
g, name, price = 9.8, "Python", 550
```

এখানে:

```
g = 9.8
name = Python
এবং price = 550 অ্যাসাইন হবে।
```

কাস্টিং

এক টাইপের ডেটা অন্য টাইপে পরিবর্তন করাকে বলে কাস্টিং। যেমন:

```
g = 9.8
y = float(11)
print(type(g))
print(type(y))
```

উপরের প্রোগ্রাম রান করলে আউটপুট দিবে:

```
<class 'float'>
<class 'float'>
```

এখানে g তে একটা ফ্লোটিং পয়েন্ট ডেটা রেখেছি, তাই ভ্যারিয়েবলটি ফ্লোট ক্লাসে রেখেছে। কিন্তু y-এর মধ্যে একটা ইন্টিজার রেখেছি। ইন্টিজার রাখার পর তাকে কাস্টিং করে বলে দিয়েছি এটিকে ফ্লোটিং পয়েন্ট হিসেবে রাখতে। তাই এটি ফ্লোট হিসেবে রেখেছে।

ডেটা ইনপুট

পাইথনে কোনো ডেটা ইনপুট নেওয়ার জন্য input() ফাংশন ব্যবহার করা হয়। যেমন:

```
name = input('Enter your name: ' )
print(name)
```

পাইথনে যে কোন কিছু স্ট্রিং আকারে ইনপুট নেয়। যদি অন্য কোন ডেটা টাইপে ডেটা ইনপুট নিতে হয়, তাহলে তাকে কাস্ট করতে হয়। যেমন:

```
age = int(input('Enter your age:'))
print(age)
```

এখানে আমরা একটা নাম্বার ইনপুট নিব। এ জন্য ইনপুট নেওয়ার পরে ইন্টিজারে এ কাস্ট করে নিতে হয়েছে।

অ্যারিথমেটিক অপারেটর

পাটি গণিতে আমরা যেসব অপারেটর ব্যবহার করেছি যেমন: যোগ, গুন, ভাগ ইত্যাদি। সেগুলোই হচ্ছে অ্যারিথমেটিক অপারেটর (Arithmetic Operators)। পাইথনের অ্যারিথমেটিক অপারেটরগুলো হচ্ছে:

অপারেটর	ব্যবহার
+	যোগ
-	বিয়োগ
*	গুন
/	ভাগ
%	মডুলাস বা রিমাইন্ডার
** ($x ** y = x^y$)	এক্সপোনেন্ট বা পাওয়ার।
//	ফ্লোর ডিভিশন

নিচের প্রোগ্রামটি দেখি:

```
x = 15
y = 4
print(x + y)
print(x - y)
print(x * y)
print(x / y)
print(x % y)
print(x ** y)
print(x // y)
```

যা আউটপুট দিবে:

```
19
11
60
3.75
3
50625
3
```

কম্পারিজন অপারেটর

কোনো কিছুর তুলনা করার জন্য কম্পারিজন (Comparison) অপারেটর ব্যবহার করা হয়। এটাকে রিলেশনাল (Relational) অপারেটরও বলা হয়।

সাধারণত দুইটা ভ্যারিয়েবলের মধ্যে এই তুলনা করা হয়। পাইথন প্রোগ্রামিংয়ের কম্পারিজন অপারেটরগুলো:

অপারেটর	অর্থ
==	সমান কি না, যাচাই করা।
!=	অসমান কি না, তা যাচাই করা
<	প্রথমটি ছোট কি না
<=	প্রথমটি ছোট অথবা সমান কি না
>	প্রথমটি বড় কি না
>=	প্রথমটি বড় অথবা সমান কি না

তুলনাগুলো যদি সত্য হয়, খালে True রিটার্ন করবে। যদি সত্য না হয়, তাহলে False রিটার্ন করবে। যেমন:

```
x = 15
y = 4
print(x == 5)
print(x != y)
print(x < y)
print(x <= 15)
print(x > y)
print(x >= y)
```