

শাফায়েতের ব্লগ

প্রোগ্রামিং ও অ্যালগরিদম টিউটোরিয়াল

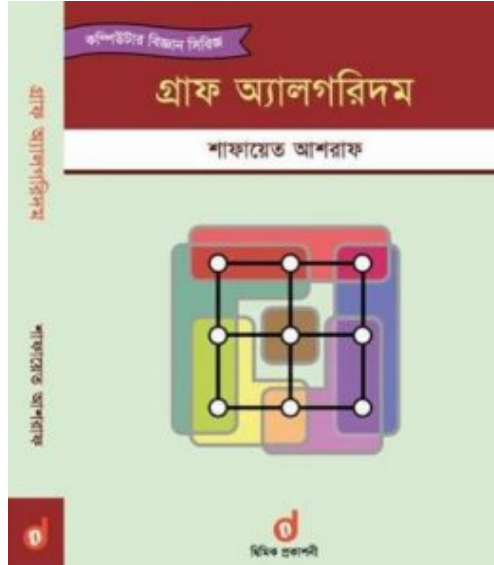
Home

অ্যালগরিদম নিয়ে যত লেখা!

আমার সম্পর্কে...

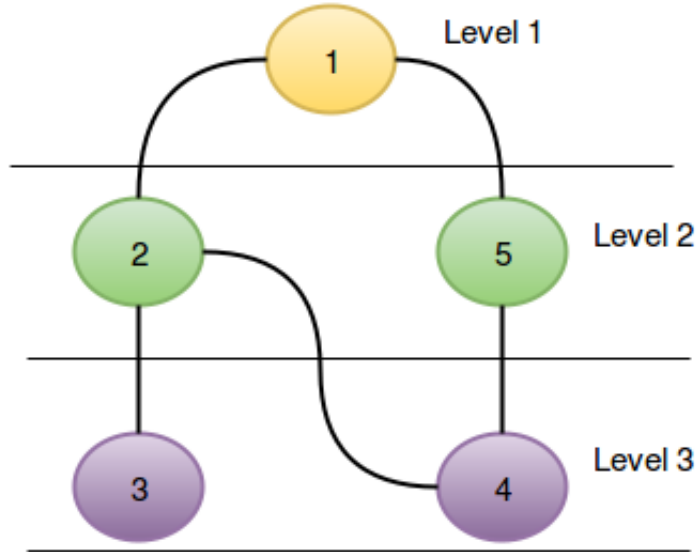
গ্রাফ থিওরিতে হাতেখড়ি ৮: ডেপথ ফার্স্ট সার্চ এবং আবারো টপোলজিকাল সর্ট

📅 মার্চ ৮, ২০১২ by শাফায়েত



আগের পর্বগুলো পড়ে থাকলে হয়তো ডেপথ ফার্স্ট সার্চ বা ডিএফএস এতদিনে নিজেই শিখে ফেলেছেন। তারপরেও এই টিউটোরিয়ালটি পড়া দরকার কিছু কনসেপ্ট জানতে।

বিএফএস এ আমরা গ্রাফটাকে লেভেল বাই লেভেল সার্চ করেছিলাম, নিচের ছবির মতো করে:



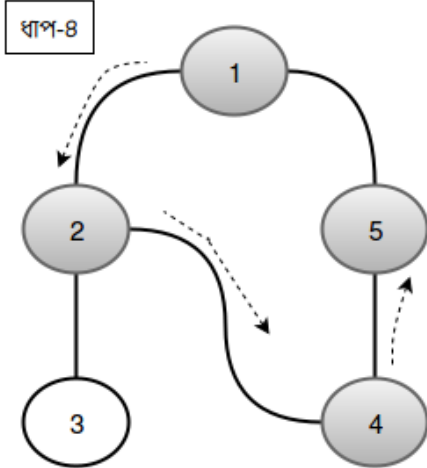
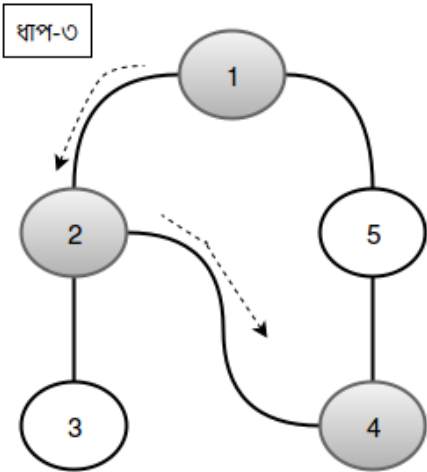
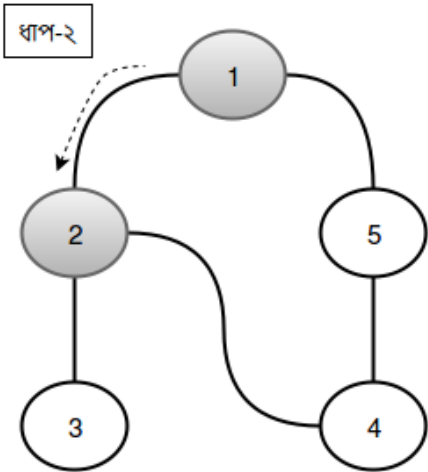
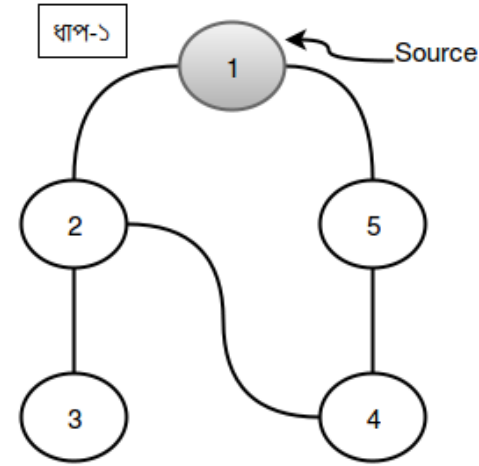
এবার আমরা কোনো নোড পেলে সাথে সাথে সে নোড থেকে আরো গভীরে চলে যেতে থাকবো, যখন আর গভীরে যাওয়া যাবেনা তখন আবার আগের নোডে ফিরে এসে অন্য আরেক দিকে যেত চেষ্টা করবো, এক নোড কখনো ২বার ভিজিট করবোনা। আমরা নোডের ৩টি রং(কালার) দিবো:

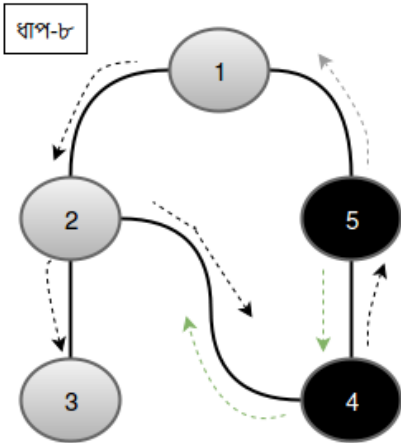
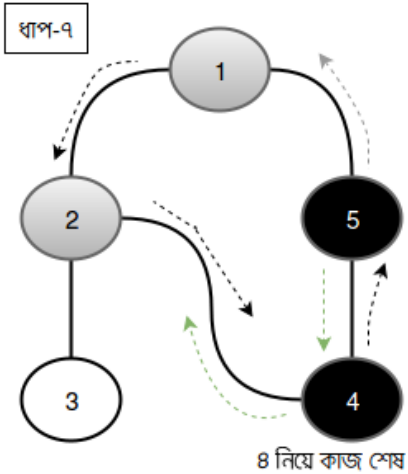
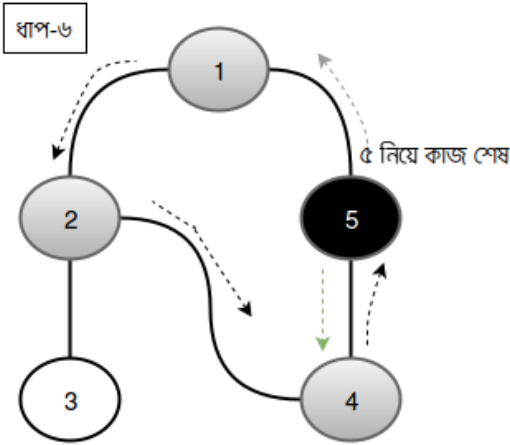
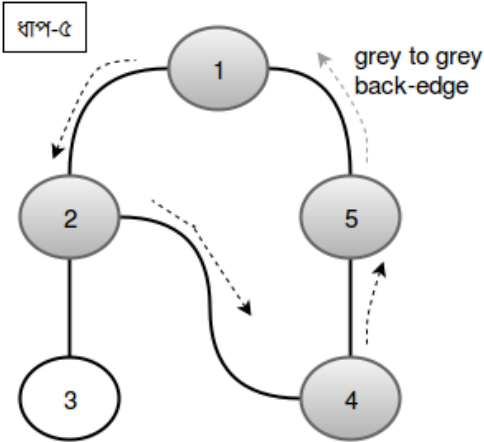
“ সাদা নোড= যে নোড এখনো খুজে পাইনি/ভিজিট করিনি।

গ্রে বা ধূসর নোড= যে নোড ভিজিট করেছি কিনুত নোডটি থেকে যেসব চাইল্ড নোডে যাওয়া যায় সেগুলো এখনো ভিজিট করে শেষ করিনি, অর্থাৎ নোডটিকে নিয়ে কাজ চলছে।

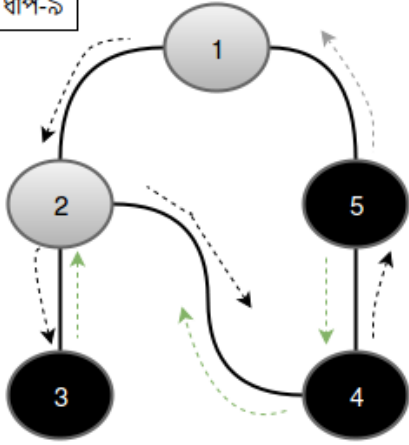
কালো নোড= যে নোডের কাজ সম্পূর্ণ শেষ।

এবার আমরা ধাপগুলো দেখতে পারি:



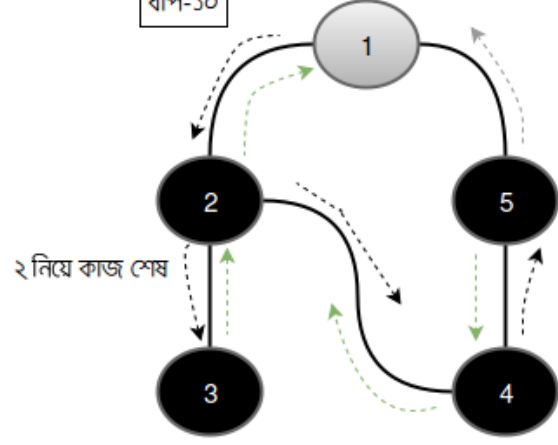


ধাপ-৯



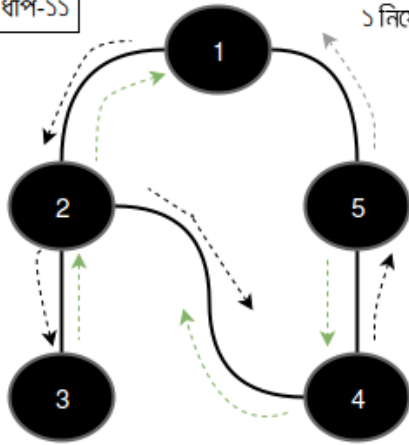
৩ নিয়ে কাজ শেষ

ধাপ-১০



২ নিয়ে কাজ শেষ

ধাপ-১১

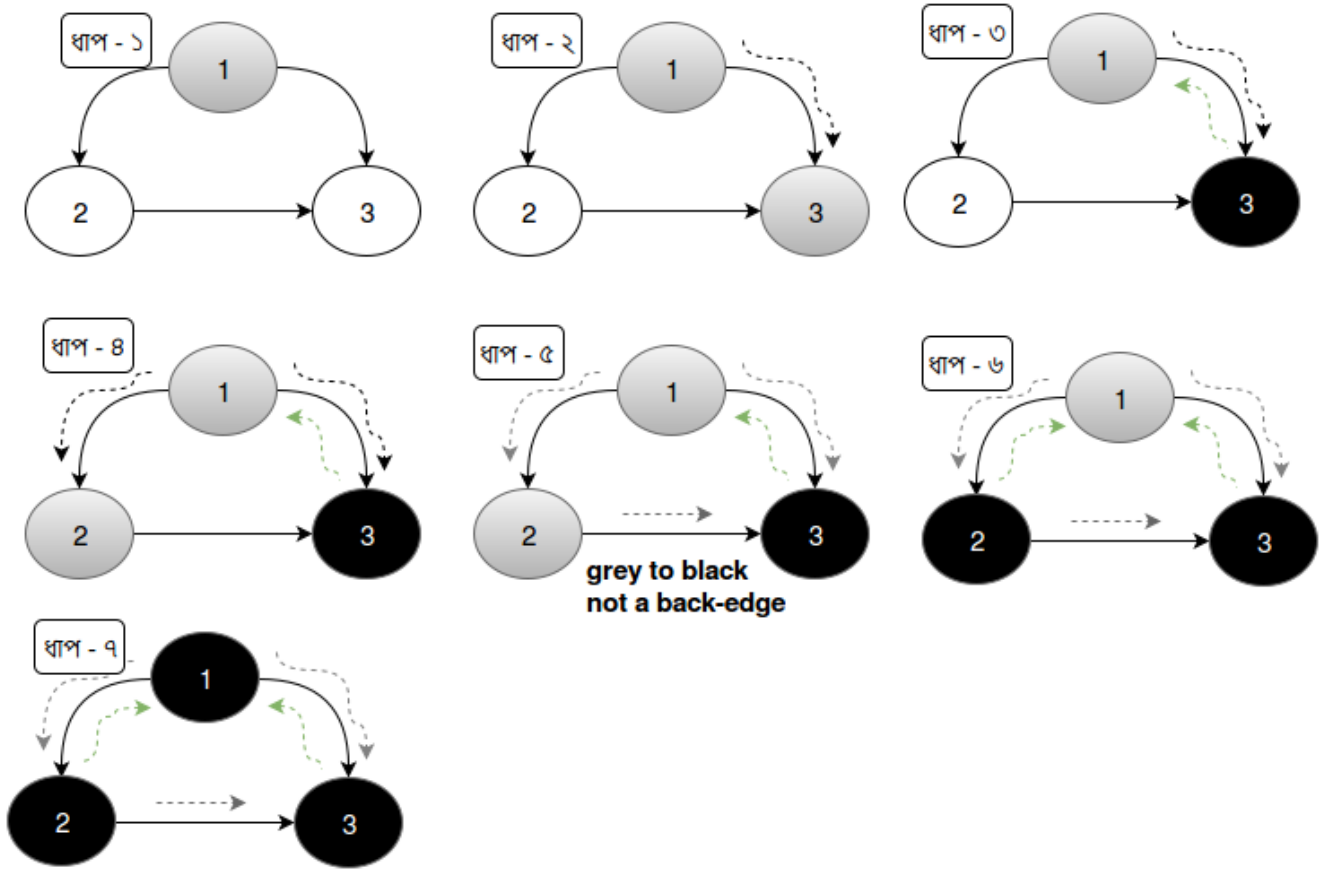


১ নিয়ে কাজ শেষ

আশা করি ডিএফএস কিভাবে কাজ করে এটা পরিষ্কার, খুব সহজ জিনিস এটা। এবার আমরা একটা খুব গুরুত্বপূর্ণ টার্ম শিখবো, সেটা হলো ব্যাকএজ(backedge)। লক্ষ করো ৫-১ কে ব্যাকএজ বলা হয়েছে। এর কারণ হলো তখনও ১ এর কাজ চলছে, ৫ থেকে ১ এ যাওয়া মানে এমন একটা নোড ফিরে যাওয়া যাকে নিয়ে কাজ এখনো শেষ হয়নি, তারমানে অবশ্যই গ্রাফে একটি সাইকেল আছে। এ ধরনের এজকে ব্যাকএজ বলে, dfs এ যদি কোনো সময় একটি গ্রে নোড থেকে আরেকটি গ্রে নোডে যেতে চেষ্টা করে তাহলে সে এজটি ব্যাকএজ এবং গ্রাফে অবশ্যই সাইকেল আছে। dfs এর সোর্স নোড এবং নোড ভিজিট করার অর্ডার এর উপর নির্ভর করে সাইকেলে যে কোনো এজকে ব্যাকএজ হিসাবে পাওয়া যেতে পারে, যেমন ১ থেকে আগে ২ এ না গিয়ে ৫ এ গেলে পরে ২-১ কে ব্যাকএজ হিসাবে পাওয়া যেতো।

আর যখন আমরা স্বাভাবিক ভাবে গ্রে থেকে সাদা নোডে যাচ্ছি তখন সে এজগুলোকে বলা হয় **ট্রি এজ**। শুধুমাত্র ট্রি এজ গুলো রেখে বাকি এজগুলো মুছে দিলে যে গ্রাফটা থাকে তাকে বলা হয় **ডিএফএস ট্রি**।

আনডিরেক্টেড গ্রাফের ক্ষেত্রে আগে ভিজিট করা কোনো নোডে ফিরে গেলেই সেটা ব্যাকএজ, কালার চেক না করলেও হয়। তবে ডিরেক্টেড গ্রাফের ক্ষেত্রে অবশ্যই করতে হবে। পরের ছবিটা দেখো:



২-৩এর এজটাকে ব্যাকএজ বলা যাচ্ছেনা, কারণ ৩ এর কাজ আগেই শেষ হয়ে গেছে।

প্রতিটা নোড আর এজ নিয়ে একবার করে করছি, dfs এর কমপ্লেক্সিটি $O(V+E)$ ।

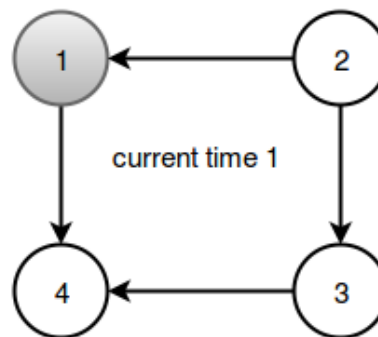
আমরা টপোলজিকাল সর্টের সমস্যা সমাধান করেছিলাম বারবার indegree উঠিয়ে। এবার আমরা খুব সহজে dfs দিয়ে এটা করবো। টপোলজিকাল কি সেটা না জানলে আগে **এই পোস্টটা পড়ো**, তারপর আগাও।

মনে করি আমাদের এজগুলো হলো: ২-১, ২-৩, ৩-১, ১-৩। অর্থাৎ ১ নম্বর কাজ করার আগে ২ নম্বরটি করতে হবে ইত্যাদি। এবার আমরা dfs চালানোর সময় একটি স্টপওয়াচ চালু করে দিবো। আর কোনো নোড নিয়ে কাজ শুরু করলে ঘড়ি দেখে নোডটি starting time/discovery time লিখে রাখবো, কাজ শেষ হলো নোডটির finishing time লিখে রাখবো।

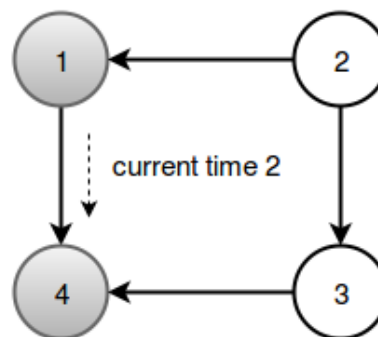
d[] = discovery time

f[] = finishing time

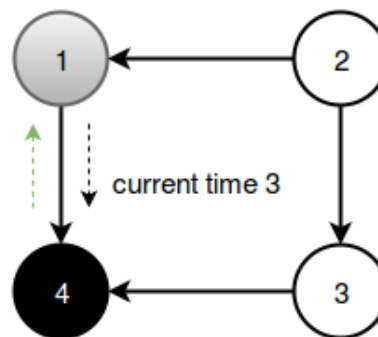
node	d[]	f[]
1	1	null
2	null	null
3	null	null
4	null	null



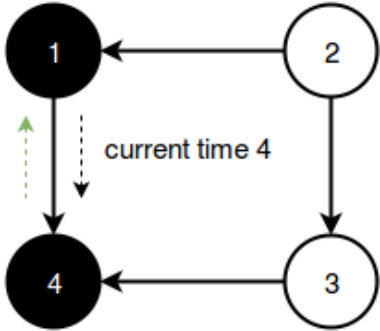
node	d[]	f[]
1	1	null
2	null	null
3	null	null
4	2	null



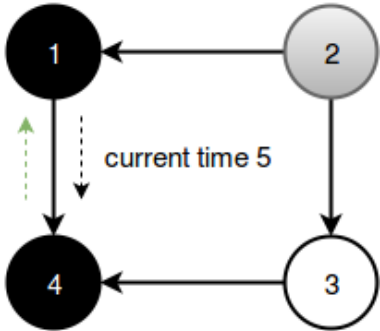
node	d[]	f[]
1	1	null
2	null	null
3	null	null
4	2	3



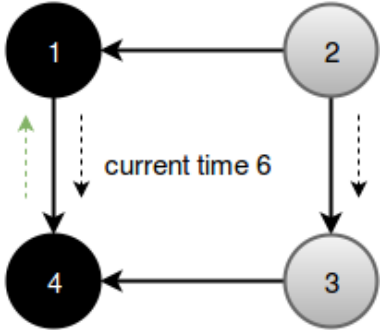
node	d[]	f[]
1	1	4
2	null	null
3	null	null
4	2	3



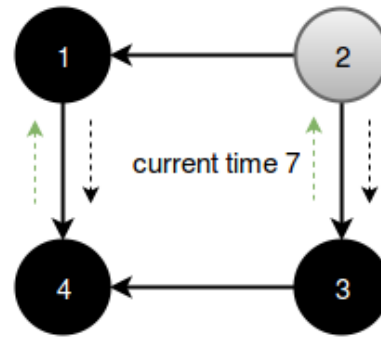
node	d[]	f[]
1	1	4
2	5	null
3	null	null
4	2	3



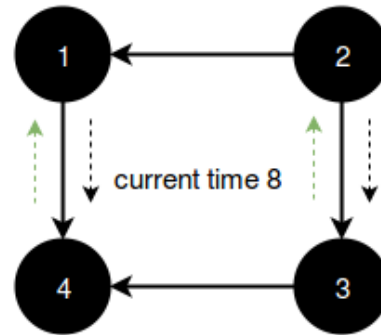
node	d[]	f[]
1	1	4
2	5	null
3	6	null
4	2	3



node	d[]	f[]
1	1	4
2	5	null
3	6	7
4	2	3



node	d[]	f[]
1	1	4
2	5	8
3	6	7
4	2	3



ফিনিশিং টাইম অনুযায়ী সর্ট করে পাই: ২,৩,১,৪

finishing time দেখে আমরা সহজেই টপসর্ট করতে পারি। যে নোডটি সবার আগে আসবে তার finishing time অবশ্যই সবথেকে বেশি হবে, কারণ প্রথম নোডের উপর নির্ভরশীল সব নোড ঘুরে আসার পরে সে নোডের finishing time assign করা হয়। **uva 11504-dominos** প্রবলেমে আগে নোডগুলোকে finishing time দিয়ে সর্ট করে তারপর আবার dfs চালাতে হয়, প্রবলেমটা চেষ্টা করো।

ডিএফএস দিয়ে আমরা যেসব কাজ করি সেগুলোর অনেকগুলোই bfs দিয়ে করতে পারি। bfs এ সাধারণত টাইম কমপ্লেক্সিটি কম হয় তবে dfs কোডিং করতে খুব কম সময় লাগে। একটা সিম্পল dfs এর সুডোকোড এরকম:

```

1 procedure DFS(G, source):
2   U ← source
3   time ← time+1
4   d[u] ← time
5   color[u] ← GREY
6   for all edges from u to v in G.adjacentEdges(v) do
7     if color[v] = WHITE
8       DFS(G,v)
9     end if
10  end for
11  color[u] ← BLACK
12  time ← time+1
13  f[u] ← time
14  return

```

নিচের প্রবলেমগুলো সলভ করতে চেষ্টা করো:

<http://uva.onlinejudge.org/external/2/280.html>

<http://uva.onlinejudge.org/external/115/11518.html>

<http://uva.onlinejudge.org/external/104/10452.html>

এরপরে এই আর্টিকেলটা পড়ে ফেলো বিস্তারিত জানার জন্য, আমার লেখা পড়ে তুমি বেসিকটা শিখতে পারবে, বিস্তারিত জানতে এবং কঠিন প্রবলেম সলভ করতে আরো অনেক কিছু জানতে হবে।

ফেসবুকে মন্তব্য

6 comments

5 Comments

Sort by **Oldest**

Add a comment...

**Mukit Chowdhury** ·

Solution Delivery Engineer at SSD-TECH

like! 1

Like · Reply · 2 · 14 April 2012 16:23

**Arifuzzaman Faisal** ·

Author at ম্যাটল্যাব পরিচিতি :: Introduction to MATLAB



Like · Reply · 1 · 4 July 2012 08:26

**Abdulla Al Sun** ·

Vice President at SGIPC

There are problems in the link of Problems (need to solve).

Like · Reply · 28 April 2013 12:15

**মোঃ মাসুদুর রহমান সাদ** ·

Junior Software Engineer at Tw inBit Limited

valo laglo

Like · Reply · 7 November 2013 14:33

**Mahade Hasan** ·

Software Engineer at BJIT Ltd.

when use dfs to find top sort

how to define which node to start dfs??????

Like · Reply · 1 · 22 November 2013 00:28

Facebook Comments Plugin

Powered by **Facebook Comments**

Posted in অ্যালগোরিদম/প্রবলেম সলভিং, প্রোগ্রামিং ? Tagged গ্রাফ থিওরি, ডিএফএস, প্রোগ্রামিং

30,097 বার পড়া হয়েছে

◀ মডুলার অ্যারিথমেটিক

ডাইনামিক প্রোগ্রামিং এ হাতেখড়ি-১(শুরুর কথা) ▶

12 thoughts on "গ্রাফ থিওরিতে হাতেখড়ি ৮: ডেপথ ফার্স্ট সার্চ এবং আবারো টপোলজিকাল সর্ট"

**ফারসান**

এপ্রিল ৩০, ২০১২ at ৩:০৭ pm

ভাইয়া শেষ animation এর ১-২ এর edge direction টা মনে হয় উলটা হবে...

Reply

**শাফায়েত**

জুলাই ২, ২০১২ at ৯:৩৩ pm

আমারতো মনে হচ্ছে ঠিকই আছে..

Reply

**মেহেদী**

ডিসেম্বর ৩০, ২০১২ at ১:৫৭ am

ভাইয়া , গ্রাফ থিওরি এর ফ্লাড-ফীল টপিক এর উপর ১ টা টীয়োটোরিয়াল লিখলে আমাদের জন্য খুবই ভালো হতো।

Reply

**Crazy Mad**

ডিসেম্বর ৩১, ২০১২ at ৪:৫৬ am

ভালো লাগলো ।তবে একটা complete কোড দিলে ভালো হত।

Reply

**SUN**

এপ্রিল ২৮, ২০১৩ at ১২:১৭ pm

There are some problems in the link of problems (need to solve)

Reply

**শাফায়েত**

এপ্রিল ২৮, ২০১৩ at ২:৪১ pm

Fixed, thanks.

Reply

**Sumit**

আগস্ট ৩, ২০১৩ at ২:০১ am

I am finding a problem on pseudo code. I think I need to increase time by 1 before recording finishing time. I am also in a confusion.

Reply

**নাবিদ**

ডিসেম্বর ৮, ২০১৩ at ১২:২৭ am

 $f[u] \leftarrow \text{time}$

এর আগে আর একবার time কে increase করতে হবে না? নাহলে discovered time আর finished time এক হয়ে যাচ্ছে?

Reply

**শাহরিয়ার**

এপ্রিল ১৩, ২০১৫ at ৯:৩৪ pm

ভাইয়া, ডিএফএস দিয়ে টপ সর্ট করার সময় যেকোনো ভারটেক্স থেকে শুরু করলেই কি হবে? আর cycle

আছে এমন গ্রাফে কি টপ সর্ট করা যায় ?

Reply

Pingback: [শাফায়েতের ব্লগ » Blog Archive](#)



TAmim

আগস্ট ৩, ২০১৬ at ৯:১৮ am

vy a dfs er ekta simple code diye bhujaile valo hoito

Reply

Pingback: [গ্রাফ থিওরিতে হাতেখড়ি ১৩: আর্টিকুলেশন পয়েন্ট এবং ব্রিজ | শাফায়েতের ব্লগ](#)

Leave a Reply

Connect with:

Powered by [OneAll Social Login](#)

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Time limit is exhausted. Please reload CAPTCHA.

8 × = 48 

Post Comment

phonetic

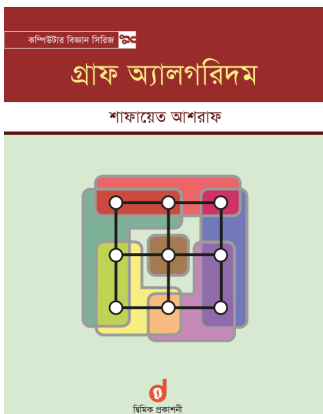
probhat

english

সাবস্ক্রাইব

Powered by [OneAll Social Login](#)

আমার সম্পর্কে

শাফায়েত, সফটওয়্যার ইঞ্জিনিয়ার @ **HACKERRANK** (বিস্তারিত...)

Like Share {2.7k}

প্রোগ্রামিং কনটেন্ট এবং অ্যালগোরিদম

অনুপ্রেরণা(৩):

কেন আমি প্রোগ্রামিং শিখবো?

কম্পিউটার বিজ্ঞান কেন পড়বো?

প্রোগ্রামিং কনটেন্ট এবং অনলাইন জাজে হাতেখড়ি

অ্যালগোরিদম বেসিক(৬):

বিগ "O" নোটেশন

কমপ্লেক্সিটি ক্লাস(P-NP, টুরিং মেশিন ইত্যাদি)

হাল্টিং প্রবলেম(নতুন)

বাইনারি সার্চ - ১

বাইনারি সার্চ - ২(বাইসেকশন)

ফ্লয়েড সাইকেল ফাইন্ডিং অ্যালগোরিদম

ডাটা স্ট্রাকচার(১১):

লিংকড লিস্ট

স্ট্যাক

কিউ+সার্কুলার কিউ(নতুন)

স্লাইডিং রেঞ্জ মিনিমাম কুয়েরি (ডিকিউ)

ডিসজয়েন্ট সেট(ইউনিয়ন ফাইন্ড)

ট্রাই(প্রিফিক্স ট্রি/রেডিক্স ট্রি)

সেগমেন্ট ট্রি-১

সেগমেন্ট ট্রি-২(লেজি প্রপাগেশন)

অ্যারে কমপ্রেসন/ম্যাপিং

লোয়েস্ট কমন অ্যানসেস্টর

বাইনারি ইনডেক্সড ট্রি

গ্রাফ থিওরি(১৮):

গ্রাফ থিওরিতে হাতেখড়ি

অ্যাডজেসেন্সি ম্যাট্রিক্স

অ্যাডজেসেন্সি লিস্ট

ব্রেথড ফার্স্ট সার্চ (বিএফএস)

মিনিমাম স্প্যানিং ট্রি ১ (প্রিমস অ্যালগোরিদম)

মিনিমাম স্প্যানিং ট্রি ২ (ক্রুসকাল অ্যালগোরিদম)

টপোলজিকাল সর্ট

ডেপথ ফার্স্ট সার্চ এবং আবারো টপোলজিকাল সর্ট

ডায়াক্সট্রা

ফ্লয়েড ওয়ার্শল

বেলম্যান ফোর্ড

আর্টিকুলেশন পয়েন্ট এবং ব্রিজ

স্ট্রংলি কানেক্টেড কম্পোনেন্ট

ম্যাক্সিমাম ফ্লো-১

ম্যাক্সিমাম ফ্লো-২

স্টেবল ম্যারেজ প্রবলেম

মিনিমাম ভারটেক্স কভার

ট্রি এর ডায়ামিটার নির্ণয়

লংগেস্ট পাথ প্রবলেম(নতুন)

অ্যালগোরিদম গেম থিওরি(৩):

গেম থিওরি-১

গেম থিওরি-২

গেম থিওরি-৩

ডাইনামিক প্রোগ্রামিং(৮):

শুরুর কথা

ডিপি 'স্টেট', NcR, ০-১ ন্যাপস্যাক

কয়েন চেঞ্জ, রক ক্লাইম্বিং

ডিপি সলিউশন প্রিন্ট করা এবং LIS

বিটমাস্ক ডিপি

মিনিমাম ভারটেক্স কভার(গ্রাফ+ডিপি)

লংগেস্ট কমন সাবসিকোয়েন্স(LCS)

ম্যাট্রিক্স চেইন মাল্টিপ্লিকেশন

ব্যাকট্র্যাকিং(১):

ব্যাকট্র্যাকিং বেসিক এবং পারমুটেশন জেনারেটর

নাস্বর থিওরি/গণিত(৪):

মডুলার অ্যারিথমেটিক

প্রাইম জেনারেটর (Sieve of Eratosthenes)

বিটওয়াইজ সিভ

ডিরেঞ্জমেন্ট

স্ট্রিং ম্যাচিং(১):

রবিন-কার্প স্ট্রিং ম্যাচিং(নতুন)

অন্যান্য(৩):

ডিরেকশন অ্যারে

মিট ইন দ্যা মিডল

কোয়ান্টাম কম্পিউটার(২)

কোয়ান্টাম কম্পিউটার কী?

কোয়ান্টাম কম্পিউটারের শক্তি এবং সীমাবদ্ধতা

AccessPress Staple | WordPress Theme: AccessPress Staple by AccessPress Themes

