গ্রাফ থিওরিতে হাতেখড়ি-৪(ব্রেডথ ফার্স্ট সার্চ)

ফেব্রুয়ারি ২২, ২০১৪ *by* শাফায়েত

আগের পর্বগুলোতে আমরা দেখেছি কিভাবে ম্যাট্রিক্স বা লিস্ট ব্যবহার করে গ্রাফ স্টোর করতে হয়। এবার আমরা প্রথম অ্যালগোরিদম দেখবো এর দিকে যাবো। শুরুতেই আমরা যে অ্যালগোরিদমটা শিখব তার নাম ব্রেডথ ফার্স্ট সার্চ(breadth first search,bfs)।

বিএফএস এর কাজ হলো গ্রাফে একটা নোড থেকে আরেকটা নোডে যাবার শর্টেস্ট পাথ বের করা। বিএফএস কাজ করবে শুধুমাত্র আন-ওয়েটেড গ্রাফের ক্ষেত্রে, তারমানে সবগুলো এজের কস্ট হবে ১।

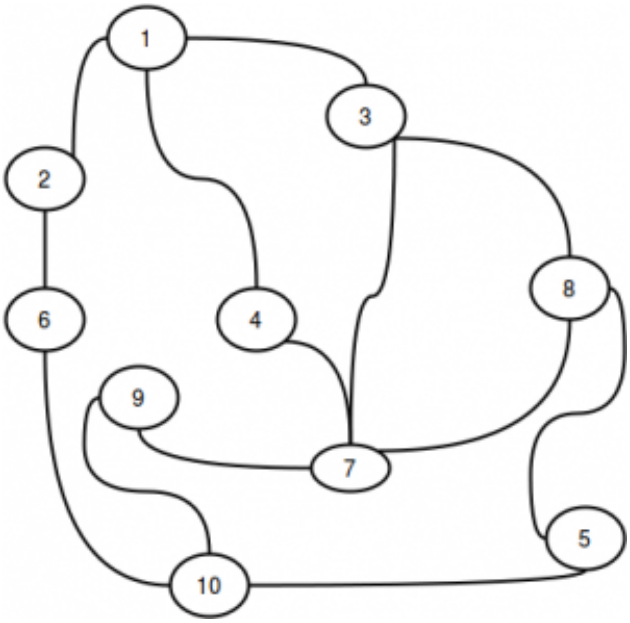
বিএফএস অ্যালগোরিদমটা কাজ করে নিচের ধারণারগুলোর উপর ভিত্তি করে:

> ১. কোনো নোডে ১ বারের বেশি যাওয়া যাবেনা
> ২. সোর্স নোড অর্থাৎ যে নোড থেকে শুরু করছি সেটা ০ নম্বর লেভেলে অবস্থিত।
> ৩. সোর্স বা ‘লেভেল ০’ নোড থেকে সরাসরি যেসব নোডে যাওয়া যায় তারা সবাই ‘লেভেল ১’ নোড।
> ৪. ‘লেভেল ১’ নোডগুলো থেকে সরাসরি যেসব নোডে যাওয়া যায় তারা

সবাই ‘লেভেল ২’ নোড। এভাবে লেভেল এক এক করে বাড়তে থাকবে।

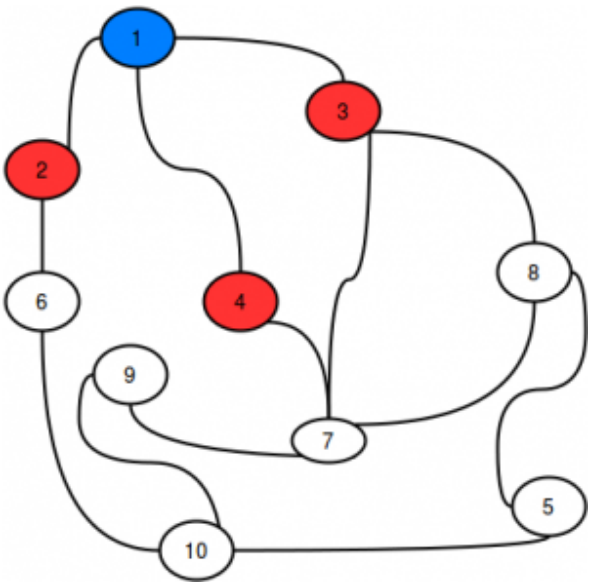
৫. যে নোড যত নম্বর লেভেলে,সোর্স থেকে তার শর্টেস্ট পথের দৈর্ঘ্য তত।

উপরে লেখাগুলো পুরোপুরি না বুঝলে আমরা একটা উদাহরণ দেখে বাকিটা পরিস্কার করব।

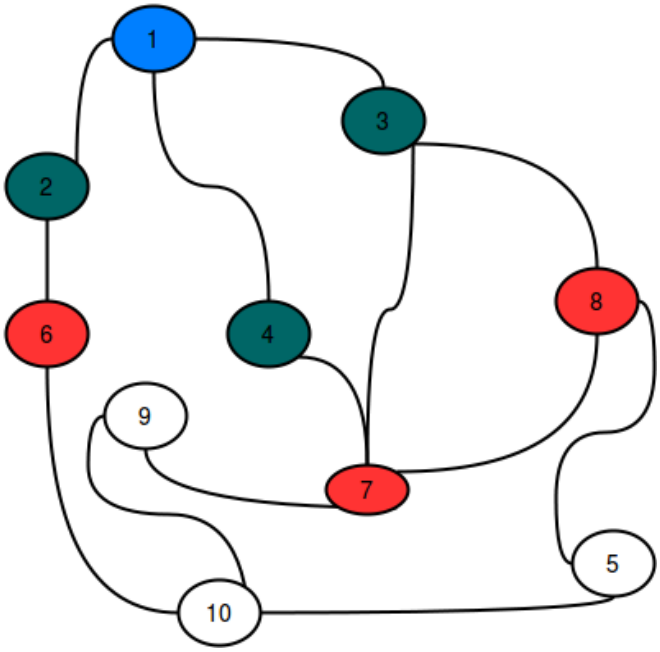


ধর তুমি ১ নম্বর শহর থেকে ১০ নম্বর শহরে যেতে চাও। প্রথমে আমরা সোর্স ধরলাম ১ নম্বর নোডকে। ১ তাহলে একটা ‘লেভেল ০’ নোড। ১ কে ভিজিটেড চিহ্নিত করি।

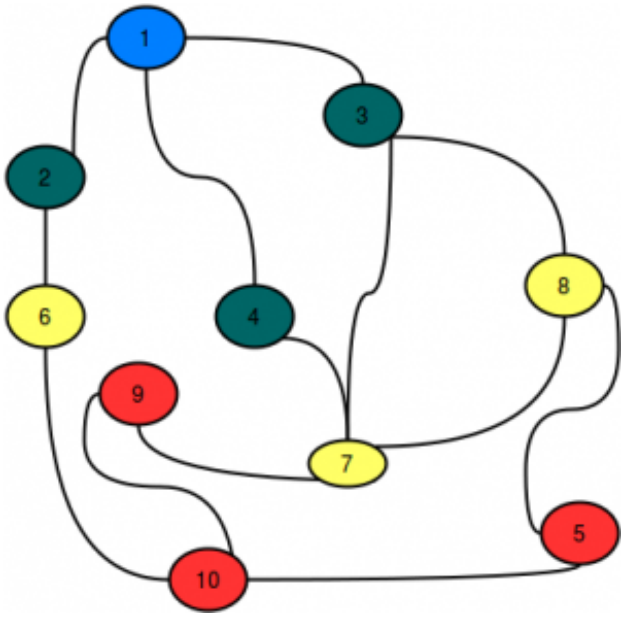
১ থেকে সরাসরি যাওয়া যায় ২,৩,৪ নম্বর নোডে। তাহলে ২,৩,৪ হলো ‘লেভেল ১’ নোড। এবার সেগুলোকে আমরা ভিজিটেড চিহ্নিত করি এবং সেগুলো নিয়ে কাজ করি। নিচের ছবি দেখ:



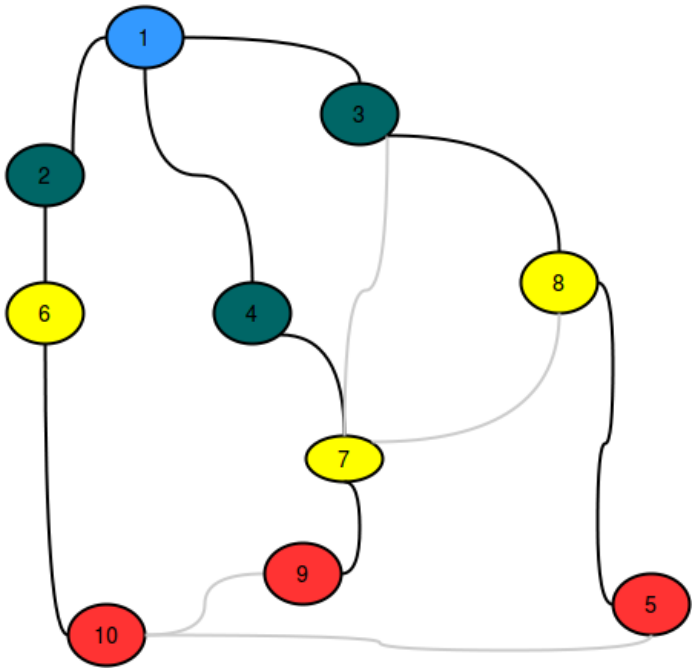
লাল নোডগুলো নিয়ে আমরা এখন কাজ করবো। রঙিন সবগুলো নোড ভিজিটেড, **এক নোডে ২বার কখনো যাবোনা**। ২,৩,৪ থেকে শর্টেস্ট পথে যাওয়া যায় ৬,৭,৮ এ। সেগুলো ভিজিটেড চিহ্নিত করি:



লক্ষ কর যে নোডকে যত নম্বর লেভেলে পাচ্ছি,সোর্স থেকে তার শর্টেস্ট পথের দৈর্ঘ্য ঠিক তত। যেমন ২নম্বর লেভেলে ৮কে পেয়েছি তাই ৮ এর দুরত্ব ২। ছবিগুলোকে একেকটা লেভেলের একেক রং দেয়া হয়েছে। আর লাল নোড দিয়ে বুঝানো হয়েছে আমরা এখন ওগুলো নিয়ে কাজ করছি। আমরা ১০ এ পৌছাইনি তাই পরের নোডগুলো ভিজিট করে ফেলি:



আমরা দেখতে পাচ্ছে ২টি লেভেল পার হয়ে ৩ নম্বর লেভেলে আমরা ১০ কে পাচ্ছি। তাহলে ১০ এর শর্টেস্ট পথ ৩। লেভেল বাই লেভেল গ্রাফটাকে সার্চ করে আমরা শর্টেস্ট পথ বের করলাম। যেসব এজ গুলো আমরা ব্যবহার করিনি সেগুলোকে বাদ দিয়ে ছবিটিকে নিচের মত করে আকতে পারি:



যেসব এজ ব্যবহার করিনি সেগুলো হালকা করে দিয়েছি,এই এজ গুলো বাদ দিলে গ্রাফটি একটি ট্রি হয়ে যায়। এই ট্রি টাকে বলা হয় বিএফএস ট্রি।

তারমানে আমাদের কাজ গুলো সোর্স থেকে লেভেল ১ নোডগুলোতে যাওয়া, তারপর লেভেল ১ এর নোডগুলো থেকে লেভেল ২ নোডগুলো খুজে বের করা, এভাবে যতক্ষন না গন্তব্যে পৌছে যাচ্ছি অথবা সব নোড ভিজিট করা শেষ হয়ে গিয়েছে ততক্ষণ কাজ চলতে থাকবে।

কিউ ডাটা স্ট্রাকচারটার সাথে আশা করি সবাই পরিচিত। কিউ হলো হুবুহু বাসের লাইনের মতো ডাটা স্ট্রাকচার। যখন একটা সংখ্যা কিউতে যোগ করা হয় তখন সেটা আগের সবগুলো সংখ্যার পিছে গিয়ে দাড়ায়, যখন কোন একটা সংখ্যা বের করে ফেলা হয় তখন সবার প্রথমের সংখ্যাটা নেয়া হয়। একে বলা ফার্স্ট ইন ফার্স্ট আউট। আমরা বিএফএস এ কিউ কাজে লাগাতে পারি। লেভেল ১ থেকে যখন কয়েকটা নতুন লেভেল ২ নোড পাবো সেগুলোকে কিউতে বা লাইনে অপেক্ষা করিয়ে রাখবো, আর সবসময় প্রথম নোডটা নিয়ে কাজ করবো। তাহলে বড় লেভেলের নোডগুলো সবসময় পিছের দিকে থাকবে, আমরা ছোট লেভেলগুলো নিয়ে কাজ করতে করতে আগাবো। উপরের গ্রাফের জন্য এটা আমরা সিমুলেট করে দেখি:

প্রথমে কিউতে সোর্স পুশ করবো:

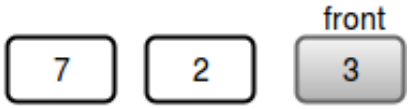
১ এর লেভেল হবে ০ বা লেভেল[১] = ০। এবার বিএফএস শুরু করবো।

প্রথমে কিউ এর সবার সামনের নোডটাকে নিয়ে কাজ করবো। সবার সামনে আছে ১, সেখান থেকে যাওয়া যায়

৪, ৩, ২ এ। ৪, ৩, ২ এ এসেছি ১ থেকে, তাহলে লেভেল[৪] = লেভেল[১] + ১ = ১, লেভেল[৩] = লেভেল[১] + ১ = ১, লেভেল[২] = লেভেল[১] + ১ = ১।

১ কে ফেল দিয়ে এদেরকে কিউতে পুশ করে রাখি:

এবার ৪ নিয়ে কাজ করি। ৪ থেকে যাওয়া যায় ৭ এ। তাহলে আমরা বলতে পারি লেভেল[৭]=লেভেল[৪]+১=২। ৪ কে ফেলে দিয়ে ৭ কে কিউতে পুশ করি:



৩ থেকে ৭,৮ এ যাওয়া যায়। ৭ কে এরই মধ্যে নিয়েছি, শুধু ৮ পুশ করতে হবে। লেভেল[৮]=লেভেল[৩]+১=২।

এভাবে যতক্ষণনা কিউ খালি হচ্ছে ততক্ষণ কাজ চলতে থাকবে। লেভেল[] অ্যারের মধ্যে আমরা পেয়ে যাবো সোর্স থেকে সবগুলো নোডের দূরত্ব!

সুডোকোড:

```
1  procedure BFS(G,source):
2      Q=queue(), level[]=infinity
3      Q.enqueue(source)
4      level[source]=0
5      while Q is not empty
6         u ← Q.pop()
7         for all edges from u to v in G.adjacentEdges(v) do
8             if level[v] = infinity:
9                       level[u]=level[v]+1;
10                     Q.enqueue(v)
11            end if
12        end for
13       end while
14.   Return distance;
```

ঠিক যেভাবে সিমুলেট করেছি সেভাবেই কোডটা লিখেছি, আশা করি বুঝতে সমস্যা হচ্ছেনা।

শুধু পাথের দৈর্ঘ্য যথেষ্ট না, পাথটাও দরকার হতে পারে। লক্ষ্য করো আমরা u থেকে v তে যাবার সময় $parent[v] = u$ করে দিচ্ছি। আমরা প্রতিটা নোডের জন্য জানি কোন নোড থেকে সেই নোডে এসেছি। তাহলে আমরা যে নোডের জন্য পাথ বের করতে চাই সেই নোড থেকে তার প্যারেন্ট নোডে যেতে থাকবো

যতক্ষণনা সোর্সে পৌছে যাই। খুবই সহজ কাজ, পাথ বের করার কোড করা তোমার উপর ছেড়ে দিলাম।

কমপ্লেক্সিটি:

প্রতিটা নোডে একবার করে গিয়েছি, প্রতিটা এজ এ একবার গিয়েছি। তাহলে কমপ্লেক্সিটি হবে $O(V+E)$ যেখানে V হলো নোড সংখ্যা এবং E হলো এজ সংখ্যা।

কখনো কখনো ২-ডি গ্রিডে বিএফএস চালানো লাগতে পারে। যেমন একটা দাবার বোর্ডে একটি ঘোড়া আর একটা রাজা আছে। ঘোড়াটা মিনিমাম কয়টা মুভে রাজার ঘরে পৌছাতে পারবে? অথবা একটা ২-ডি অ্যারেতে কিছু সেল ব্লক করে দেয়া হয়েছে, এখন কোনো সেল থেকে আরেকটি সেলে মিনিমাম মুভে পৌছাতে হবে, প্রতি মুভে শুধুমাত্র সামনে-পিছে-বামে-ডানে যাওয়া যায়। আগে নোডকে আমরা প্রকাশ করছিলাম একটা মাত্র সংখ্যা দিয়ে, এখন নোডকে প্রকাশ করতে হবে দুটি সংখ্যা দিয়ে, রো(row) নাম্বার, এবং কলাম নাম্বার। তাহলে আমরা নোড রিপ্রেজেন্ট করার জন্য সি তে একটা স্ট্রাকচার বানিয়ে নিতে পারি এরকম:

> *struct node{int r,c;};*

অথবা আমরা সি++ এর "পেয়ার" ব্যবহার করতে পারি।

> *pair<int,int>*

এ ক্ষেত্রে ভিজিটেড, প্যারেন্ট, লেভেল অ্যারেগুলো হবে ২ ডিমেনশনের, যেমন $visited[10][10]$ ইত্যাদি। কিউতে নোডের বদলে স্ট্রাকচার পুশ করবো। আর কোন একটা ঘর থেকে অন্য ঘরে যাবার সময় চেক করতে হবে বোর্ডের বাইরে চলে যাচ্ছে কিনা। একটা স্যাম্পল সি++ কোড দেখি:

```
#define pii pair<int,int>
int fx[]={1,-1,0,0}; //ডিরেকশন অ্যারে
int fy[]={0,0,1,-1};
int cell[100][100]; //cell[x][y] যদি -১ হয় তাহলে সেলটা ব্লক
int d[100][100],vis[100][100]; //d means destination from source.
int row,col;
void bfs(int sx,int sy) //Source node is in [sx][sy] cell.
{
    memset(vis,0,sizeof vis);
    vis[sx][sy]=1;
    queue<pii>q; //A queue containing STL pairs
    q.push(pii(sx,sy));
    while(!q.empty())
    {
        pii top=q.front(); q.pop();
        for(int k=0;k<4;k++)
        {
            int tx=top.uu+fx[k];
            int ty=top.vv+fy[k]; //Neighbor cell [tx][ty]
            if(tx>=0 and tx<row and ty>=0 and ty<col and cell[tx][ty]!=-1 and vis[tx][ty]==0
            {
                vis[tx][ty]=1;
                d[tx][ty]=d[top.uu][top.vv]+1;
```

```
24                 q.push(pii(tx,ty)); //Pushing a new pair in the queue
25             }
26         }
27     }
28 }
```

তুমি যদি ডিরেকশন অ্যারের ব্যাপারটা না বুঝো তাহলে এই লেখাটা পড়লে আরো কিছু ডিটেইলস জানতে পারবে।

বিএফএস শুধুমাত্র আন-ওয়েটেড গ্রাফে কাজ করে, ওয়েটেড গ্রাফে শর্টেস্ট পাথ বের করতে ডায়াক্সট্রা অ্যালগোরিদম ব্যবহার করতে পারো। গ্রাফে নেগেটিভ সাইকেল থাকলে বেলম্যান ফোর্ড ব্যবহার করতে হবে।

প্র্যাকটিসের জন্য প্রবলেম:

Bicoloring(Bipartite checking)

A Node Too Far(Shortest path)

Risk(Shortest path)

Bombs! NO they are Mines!!(bfs in 2d grid)

Knight Moves(bfs in 2d grid)

We Ship Cheap(Printing path)

Word Transformation(strings)

পরের পর্ব

ফেসবুকে মন্তব্য

13 comments

12 Comments Sort by **Oldest**

Add a comment...

Anjan Biswas ·
BUBT - Bangladesh University of Business & Technology
এটা পড়ে আমি নিজে নিজে bicoloring solve করলাম। thanks শাফায়েত via.
আর http://e-maxx.ru/algo/bfs এ থেকে অল্প help নিসি।

Like · Reply · 8 · 15 April 2013 17:53

Kawsar Ahmed ·


Assistant Lecturer at Gono Bishw abidyalay

safaet vai valo laglo.....

Like · Reply · 8 March 2014 04:30


Shams Wadud Abbir ·
Khulna University

Safaet bhai ami ki vabe array compression , directional array er moton aro bivinno programming technique kon website a gele sikte parbo altu janaben . Janale upokrito hobo . Ar apnar tutorial deke ami graph er problem solve korte partesi . Thanks .

Like · Reply · 1 · 17 March 2014 23:04


Md Razon Hossain

at line 15, int v=G[u][i] bolte kon node k bujhacce?

Like · Reply · 1 · 9 May 2014 12:22


Our Age ·
Millionaire city

Nice ! Thank You.

Like · Reply · 22 January 2015 14:01


Adu Nana ·
Daffodil International University - DIU

at line 15, int v=G[u][i] bolte kon node k bujhacce?

Like · Reply · 6 February 2015 01:07


Johurul Islam ·
Sylhet Engineering College

allah shafaet saheb k shohih jibon dhan korun.(amin)

Like · Reply · 5 · 17 May 2015 23:03

MD Muntaz ·
Computer Science & Engineering in rajshahi university at Rajshahi

vai bhujte paritesi but problem solve korte pari na

Like · Reply · 1 March 2016 21:45


Ahsan Sadeeb ·
Ahsanullah University of Science & Technology (AUST)

what does top.uu & top.vv means in the 18th and the 19th line?

Like · Reply · 16 April 2016 08:52


S M Abu Raihan ·
Pabna University of Science & Technology, Pabna, Bangladesh.

pair er 1st element & 2nd element respectively

Like · Reply · 1 · 15 May 2016 07:52

Mahmud Hossain ·
Studying Computer Science and Engineering at Bangladesh University of Business and Technology

Why লেভেল[২]=লেভেল[৩]+১=১ ???
Isn't it should be লেভেল[২]=লেভেল[১]+১=১???.

Like · Reply · 1 · 1 May 2016 23:50

Load 2 more comments

Facebook Comments Plugin

Powered by Facebook Comments

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

54,583 বার পড়া হয়েছে

মিট ইন দ্যা মিডল টেকনিক

লোয়েস্ট কমন অ্যানসেস্টর

22 thoughts on "গ্রাফ থিওরিতে হাতেখড়ি-৪(ব্রেডথ ফার্স্ট সার্চ)"

?জাকির!

আগস্ট ২৭, ২০১১ *at* ৪:২৩ *am*

অনেক অনেক ধন্যবাদ শাফায়েত ভাই।

Reply

শাফায়েত

আগস্ট ২৮, ২০১১ *at* ২:১৭ *pm*

আপনাকেও ধন্যবাদ

Reply

Pingback: গ্রাফ থিওরিতে হাতেখড়ি-৫(কোডিং বিএফএস) | ACMSolver - Bangla

Pingback: প্রোগ্রামার শাফায়েতের যত গ্রাফ থিওরীর টিউটোরিয়াল | ACMSolver - Bangla

H@RUN

আগস্ট ১, ২০১৩ *at* ১১:০১ *am*

ভালো লাগলো

Reply

Bellal

নভেম্বর ১০, ২০১৩ *at* ৯:৩১ *am*

many many thanks.....

Reply

Shohag

জানুয়ারি ৯, ২০১৪ *at* ৯:৪১ *pm*

many many thanks.

Reply

Tuhin

ফেব্রুয়ারি ৭, ২০১৪ *at* ৩:২৪ *pm*

It is really helpful.

Thank you

Reply

Mehrab

ফেব্রুয়ারি ২৪, ২০১৪ *at* ৪:৪২ *pm*

In line 20, shouldn't it be vis[tx][ty]==0 instead of vis[sx][sy]==0?

Reply

শাফায়েত

ফেব্রুয়ারি ২৪, ২০১৪ *at* ১০:২৭ *pm*

ঠিক করে দিলাম, অনেক ধন্যবাদ। লাইন ২২ এও একই ভুল ছিলো।

Reply

Verenda, Garamond

মার্চ ২০, ২০১৪ *at* ৭:২৮ *am*

শাফায়েত ভাই,

কিউঃ[১]

লেভেল[৩]=লেভেল[১]+১=১ কেন?

লেভেল[৩]=লেভেল[২]+১=১ হওয়ার কথাছিল না?

Reply

শাফায়েত

মার্চ ২৫, ২০১৪ *at* ২:৩০ *pm*

৩ এ এসেছি ১ নাম্বার নোড থেকে। তাই ৩ এর লেভেল হবে ১ এর লেভেল এর থেকে ১ বেশি।

Reply

Razon

মে ৯, ২০১৪ *at* ১২:২৮ *pm*

at line 15, int v=G[u][i] বলতে কোন node কে বুঝানো হয়েছে ..???

Reply

Tanvir Ahmed

ডিসেম্বর ২৭, ২০১৪ *at* ১১:৩০ *am*

yes. now I understand this. it is awesome. thanks

Reply

Shafin Islam Ohin

ফেব্রুয়ারি ১২, ২০১৫ *at* ৪:০৬ *pm*

top.uu এবং top.vv কি ?

Reply

শাফায়েত

ফেব্রুয়ারি ২১, ২০১৫ *at* ১১:৩৪ *pm*

top.first, top.second

Reply

Jahir

এপ্রিল ১, ২০১৫ *at* ১২:৩৩ *am*

d[tx][ty]=d[top.uu][top.vv]+1; পরের অংশটা বুঝি নাই। d[tx][ty] মানে নোডের নতুন পজিশন এটা বুঝতে পেরেছি, কিন্তু d[top.uu][top.vv]+1 এটা কিভাবে নোডের নতুন পজিশনের সোর্স স্টোর করে সেটা বুঝতে পারি নি।

Reply

আকাশ

মে ১, ২০১৫ *at* ৩:২১ *pm*

ভাইয়া, কিউ।[১] এর শেষ লাইন লেভেল[৪]=লেভেল[৩]+১=১। এদেরকে কিউতে পুশ করে রাখি: টা বুঝিনি । যেহেতু ১থেকে এসেছে। তাই এটা কি এরকম হবে না লেভেল[৪]=লেভেল[১]+১=১ ?????

Reply

এস আর জাবেদ

জানুয়ারি ১২, ২০১৬ *at* ৩:০০ *am*

শাফায়েত ভাই অনেক ধন্যবাদ এই রকম একটা ব্লগের জন্য।আগে এক সময় এই বল্গের কিছুই বুজতাম না, এখন আসতে আসতে অনেক কিছুই ক্লিয়ার হচ্ছে ,বুজতেও পারছি।

Reply

Pingback: শাফায়েতের ব্লগ » Blog Archive

Pingback: ডাটা স্ট্রাকচার: কিউ এবং সার্কুলার কিউ | শাফায়েতের ব্লগ

Pingback: গ্রাফ থিওরিতে হাতেখড়ি ৮: ডেপথ ফার্স্ট সার্চ এবং আবারো টপোলোজিকাল সর্ট | শাফায়েতের ব্লগ

Leave a Reply

Connect with:

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

Comment

Name *

Email *

Website

Time limit is exhausted. Please reload CAPTCHA.

– 7

= 0

Post Comment

phonetic probhat english

সাবস্ক্রাইব

আমার সম্পর্কে

শাফায়েত, সফটওয়্যার ইঞ্জিনিয়ার @ HackerRank (বিস্তারিত...)

প্রোগ্রামিং কনটেস্ট এবং অ্যালগোরিদম

অনুপ্রেরণা(৩):

কেন আমি প্রোগ্রামিং শিখবো?

কম্পিউটার বিজ্ঞান কেন পড়বো?

প্রোগ্রামিং কনটেস্ট 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 কমপ্রেশন/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 টপোলোজিকাল 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):
ব্যকট্র্যাকিং বেসিক এবং পারমুটেশন জেনারেটর

নাম্বার 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