

শাফায়েতের ব্লগ

গ্রাফ থিওরিতে হাতেখড়ি ৯: ডেপথ ফার্স্ট সার্চ এবং আবারো টপোলজিকাল সর্ট

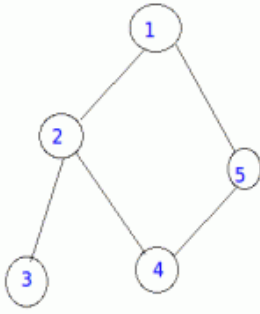
২০১২-০৩-০৮ ১৫:০৩:২৪ শাফায়েত

বেশ কিছুদিন পর **সিরিজের** পরের পর্ব লিখতে বসলাম। গ্রাফের টিউটোরিয়াল লিখতে অনেক ছবি আকতে হয় দেখে অনেক সময় চলে যায়। এখন থেকে স্থির ছবি ব্যবহার না করে gif অ্যানিমেশন ব্যবহার করবো বলে ঠিক করেছি, এতে বুঝতে সুবিধা হবে, অনেক কম কথা বলতে হবে। এটা এই সিরিজের প্রথম অ্যানিমেটেড টিউটোরিয়াল। (তবে তুমি যদি লেখার উপরের আইকনে চাপ দিয়ে pdf হিসাবে ডাউনলোড করো তাহলে অ্যানিমেশন কাজ করবে না)।

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

bfs এ আমরা গ্রাফটাকে লেভেল বাই লেভেল সার্চ করেছিলাম, নিচের অ্যানিমেশনের মতো করে:

BFS



Created by: Shafaet, university of Dhaka
Gif courtesy: <http://www.createagif.net/>

(যদি অ্যানিমেশন দেখতে সমস্যা হয় তাহলে সবগুলো ছবি ডাউনলোড করতে পারো [এখান থেকে](#))

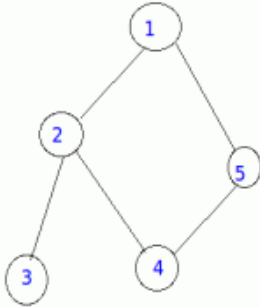
এবার আমরা কোনো নোড পেলো সাথে সাথে সে নোড থেকে আরো গভীরে চলে যেতে থাকবো, যখন আর গভীরে যাওয়া যাবে না তখন আবার আগের নোডে ফিরে এসে অন্য আরেক দিকে যেত চেষ্টা করবো, এক নোড কখনো ২বার ভিজিট করবোনা। আমরা নোডের ০টি রং(কালার) দিবো:

সাদা নোড= যে নোড এখনো খুঁজে পাইনি/ভিজিট করিনি।

গ্রে বা ধূসর নোড= যে নোড ভিজিট করেছি কিন্তু 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 অ্যানিমেশন দেখতে পারি:

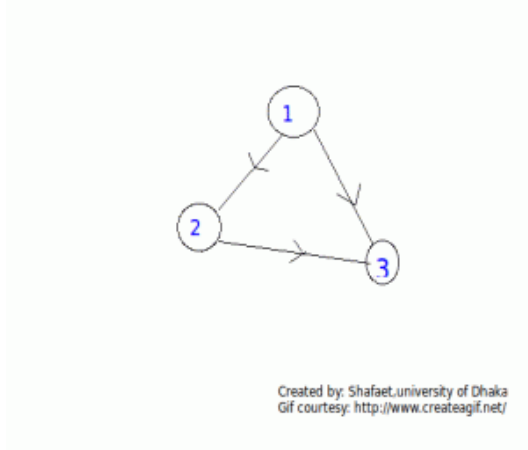


Created by: Shafaet, university of Dhaka
Gif courtesy: <http://www.createagif.net/>

আশা করি ডিএফএস কিভাবে কাজ করে এটা পরিষ্কার, খুব সহজ জিনিস এটা। এবার আমরা একটা খুব গুরুত্বপূর্ণ টার্ম শিখবো, সেটা হলো ব্যাকএজ(backedge)। অ্যানিমেশনে 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 adjacency list 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।

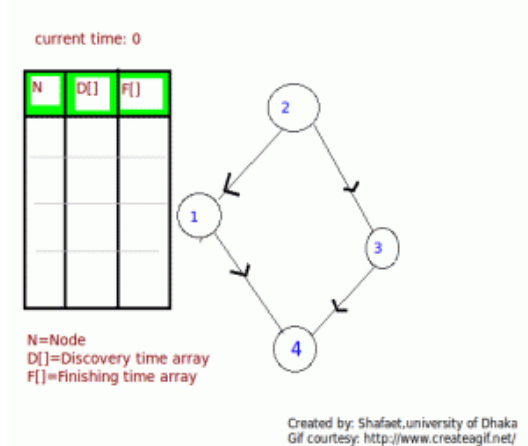
আনডিরেক্টেড গ্রাফের ক্ষেত্রে আগে ডিজিট করা কোনো নোডে ফিরে গেলেই সেটা ব্যকএজ, কালার চেক না করলেও হয়। তবে ডিরেক্টেড গ্রাফের ক্ষেত্রে অবশ্যই করতে হবে। আরেকটি ছোট অ্যানিমেশন:



২-৩এর এজটাকে ব্যাকএজ বলা যাচ্ছেনা, কারণ ৩ এর কাজ আগেই শেষ হয়ে গেছে। 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, ৩-৪, ১-৪। 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।



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:

DFS (V, E)

```
for each vertex u in V[G]
do color[u] ← WHITE
    n[u] ← NIL           ▷ n[] is parent array
time ← 0
for each vertex u in V[G]
do if color[u] ← WHITE
    then DFS-Visit(u)    ▷ build a new DFS-tree from u
```

DFS-Visit (u)

```

color[u] ← GRAY                                ▷ discover u
time ← time + 1                                ▷ stopwatch
d[u] ← time
for each vertex v adjacent to u                ▷ explore (u, v)
    do if color[v] ← WHITE
        then π[v] ← u                          ▷ saving parent to print path
        DFS-Visit(v)
color[u] ← BLACK

f[u] ← time                                    ▷ we are done with u
#http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/GraphAlgor/depthSearch.htm

```

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>

যদি অ্যানিমেশন দেখতে সমস্যা হয় তাহলে সবগুলো ছবি ডাউনলোড করতে পারো [এখান থেকে](#) ।


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।

গ্রাফ থিওরিতে হাতেখড়ি সিরিজটা কেমন লেগেছে?

- ☐ খুবই ভালো, আমি এই সিরিজ পড়েই গ্রাফ থিওরি প্রথম শিখেছি/শিখছি।
- ☐ আমি গ্রাফ থিওরি সম্পর্কে জানি, তাও এই সিরিজটি পড়ে উপকার হয়েছে।
- ☐ লেখাগুলো মোটামুটি চলনসই, তবে আরো বিস্তারিত লেখা উচিত ছিলো।
- ☐ পুরো অকাজের সিরিজ, কিছুই শিখতে পারিনি এটা থেকে।

Vote

[View Results](#)

 Loading ...

ফেসবুকে মন্তব্য

comments

Powered by [Facebook Comments](#)