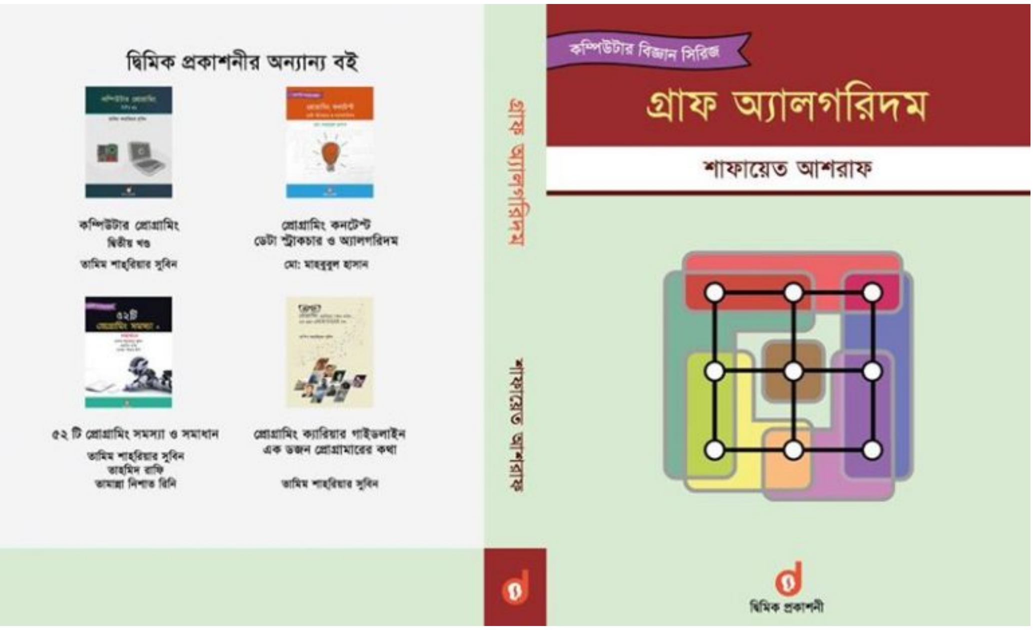
শাফায়েতের ব্লগ

প্রোগ্রামিং ও অ্যালগরিদম টিউটোরিয়াল

গ্রাফ থিওরিতে হাতেখড়ি ৬: মিনিমাম স্প্যানিং ট্রি(ক্রুসকাল অ্যালগোরিদম)

সেপ্টেম্বর ২৯, ২০১১ *by* শাফায়েত



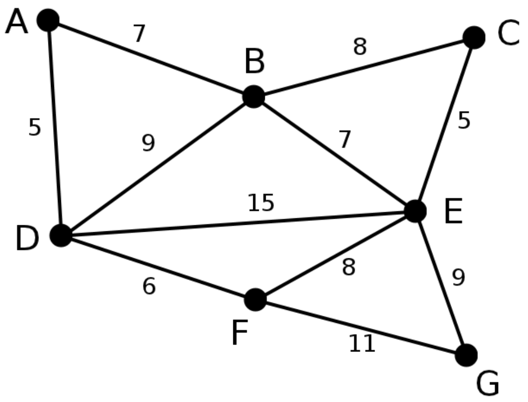
(অন্যান্য পোস্ট)

আগের পোস্টে আমরা প্রিমস অ্যালগোরিদম ব্যবহার করে mst নির্ণয় করা দেখেছি। mst কাকে বলে সেটাও আগের পোস্টে বলা হয়েছে। এ পোস্টে আমরা দেখবো mst বের করার আরেকটি অ্যালগোরিদম যা

ক্রুসকালের অ্যালগোরিদম নামে পরিচিত। এটি mst বের করার সবথেকে সহজ অ্যালগোরিদম। তবে তোমাকে অবশ্যই ডিসজয়েন্ট সেট ডাটা স্ট্রাকচার সম্পর্কে জানতে হবে,না জানলে এই পোস্টটি অবশ্যই দেখে আসো।

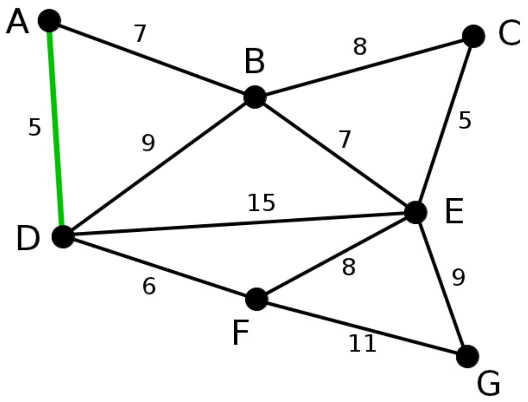
এই পোস্টে নিজের আকা ছবি ব্যবহার করবোনা। উইকিতে ক্রুসকাল নিয়ে খুব সুন্দর করে লেখা আছে,আমি ওখানকার ছবিগুলোই ব্যবহার করে সংক্ষেপে অ্যালগোরিদমটা বুঝানোর চেষ্টা করবো।

নিচের গ্রাফটি দেখো:

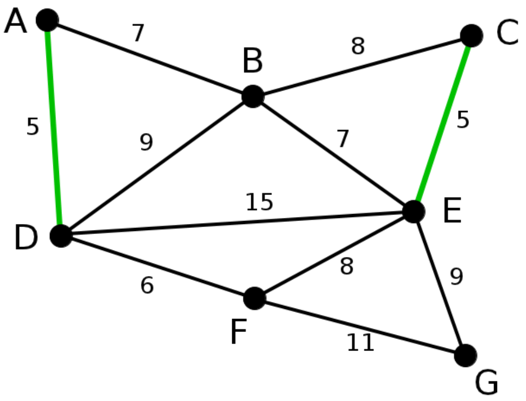


প্রথমে আমাদের ট্রিতে একটি এজও নেই। আমরা মুল গ্রাফের এজগুলোকে cost অনুযায়ী সর্ট করে ফেলবো। সব থেকে কম cost এর এজ আগে নিবো,বেশি cost এর এজ পরে নিবো। দুটি এজের cost সমান হলে যেকোনো একটি আগে নিতে পারি। তারপর একটি করে এজ নিবো আর দেখবো এজের দু প্রান্তের নোডগুলোর মধ্যে ইতোমধ্যে কোনো পথ আছে নাকি,যদি থাকে তাহলে এজটি নিলে সাইকেল তৈরি হবে,তাই এজটা আমরা নিবোনা। বুঝতেই পারছো প্রিমসের মত এটিও একটি 'গ্রিডি' অ্যালগোরিদম।

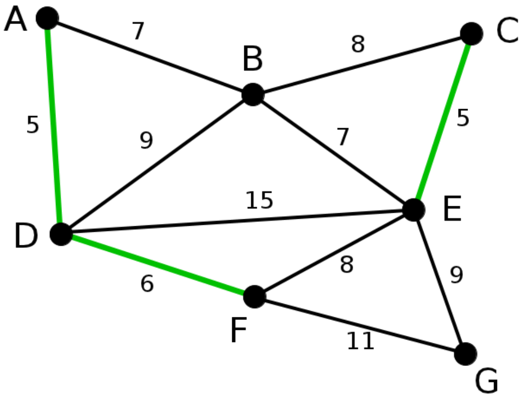
উপরে AD আর CE হলো সবথেকে কম cost এর এজ। আমরা AD কে সাবগ্রাফের অন্তর্ভূক্ত করলাম।

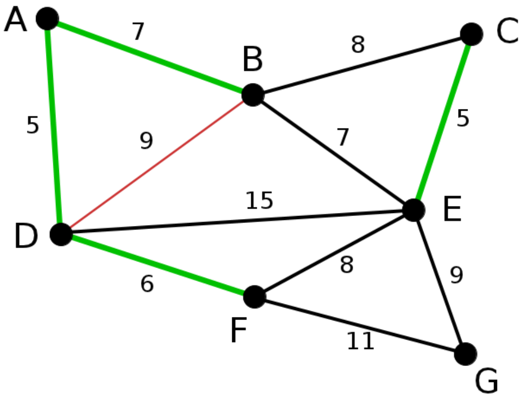


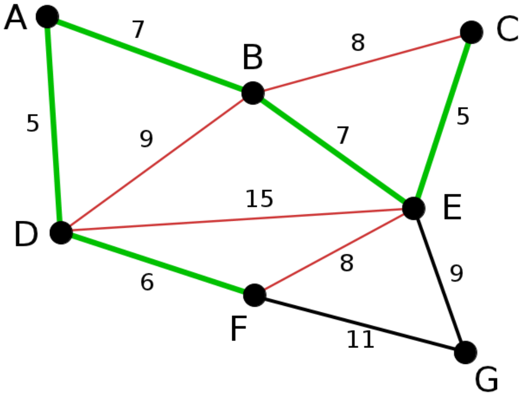
একই ভাবে এরপ CE তারপর DF,AB এবং BE কে যোগ করবো:


A
7
B
8
C
5
9
7
5
15
E
D
8
9
6
F
11
G

top

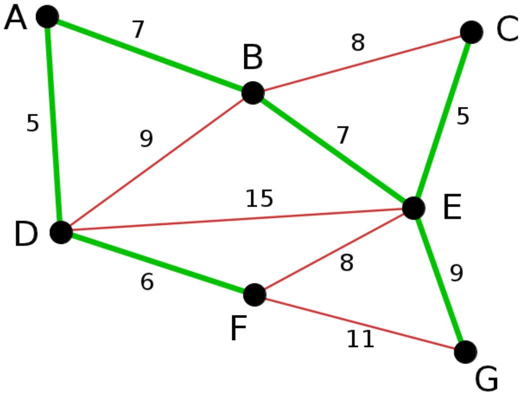

A
7
B
8
C
5
9
7
5
15
E
D
8
9
6
F
11
G


A
7
B
8
C
5
9
7
5
15
E
D
8
9
6
F
11
G



এরপর সবথেকে ছোট এজ হলো EF, এটাকে আমরা নিতে পারবোনা কারণ EF নিলে একটি সাইকেল তৈরি হয়ে যাবে,E থেকে F তে যাবার রাস্তা আগে থেকেই আছে,তাই এজটি নেয়ার কোনো দরকার নেই। এভাবে BC,DB সহ লাল রঙের এজগুলো বাদ পড়বে কারণ এরা সাইকেল তৈরি করে।

সবশেষে EG যোগ করলে আমরা mst পেয়ে যাবো।



এখন আমরা ইম্প্লিমেনটেশনে আসি। আমাদের প্রথম কাজ হলো সর্ট করা। পরের কাজ হলো একটি একটি এজ নিয়ে চেক করা যে দু প্রান্তের নোড দুটির মধ্য পথ আছে নাকি,অর্থাত তারা একই কম্পোনেন্টের ভিতর আছে নাকি। এটা চেক করতে লাগবে ডিসজয়েন্ট সেট। ডিসজয়েন্ট সেট নিয়ে টিউটোরিয়ালে দেখিয়েছিলাম কিভাবে দুটি নোড একই সাবগ্রাফে আছে নাকি বের করতে হয়। তুমি সেই কাজটিই এখানে করবে। তারপর একই সাবগ্রাফে না থাকলে আগের মত Union ফাংশন কল দিয়ে তাদের একসাথে নিয়ে আসবে আর এজটি একটি ভেক্টর বা অ্যারেতে সেভ করে রাখবে।

নিচে একটা ইমপ্লিমেন্টেশন দিলাম, আশা করি এটা কপি না করে নিজে বুঝে লিখবে:

```
struct edge {
    int u, v, w;
    bool operator<(const edge& p) const
    {
```

```
        return w < p.w;
    }
};
int pr[MAXN];
vector<edge> e;
int find(int r)
{
    return (pr[r] == r) ? r : find(pr[r]);
}
int mst(int n)
{
    sort(e.begin(), e.end());
    for (int i = 1; i <= n; i++)
        pr[i] = i;

    int count = 0, s = 0;
    for (int i = 0; i < (int)e.size(); i++) {
        int u = find(e[i].u);
        int v = find(e[i].v);
        if (u != v) {
            pr[u] = v;
            count++;
            s += e[i].w;
            if (count == n - 1)
                break;
        }
    }
    return s;
}

int main()
{
    // READ("in");
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= m; i++) {
        int u, v, w;
        cin >> u >> v >> w;
        edge get;
        get.u = u;
        get.v = v;
        get.w = w;
        e.push_back(get);
    }
    cout << mst(n) << endl;
    return 0;
}
```

কমপ্লেক্সিটি অ্যানালাইসিস:

মনে করি E হলো এজ সংখ্যা। এজগুলোকে সর্ট করতে হবে, সেটার কমপ্লেক্সিটি $O(ElogE)$, এরপরে শুধু এজগুলোর উপর লিনিয়ার লুপ চালাতে হবে। তাহলে মোট কমপ্লেক্সিটি $O(ElogE)$।

mst সম্পর্কিত অনেকগুলো সহজ প্রবলেম দিয়েছি প্রিমস এর টিউটোরিয়ালে,ওগুলো সলভ করে প্র্যাকটিস করতে পারো। আরেকটু ভালো প্রবলেম করতে চাইলে দেখো:

২য় সেরা স্প্যানিং ট্রি?

অনেক সময় প্রবলেমে বলা হয় সেকেন্ড বেস্ট MST বের করতে। এটা আমরা ব্রুট ফোর্স দিয়ে বের করতে পারি। MST বের করা পর যে এজগুলো পাবো সেগুলার প্রত্যেকটা একবার করে বাদ দিয়ে নতুন করে MST বের করতে হবে, এভাবে করে যে MST টা মিনিমাম হবে সেটাই সেকেন্ড বেস্ট MST।

http://uva.onlinejudge.org/external/103/10369.html
http://uva.onlinejudge.org/external/117/11733.html

ফেসবুকে মন্তব্য

11 comments

5 Comments Sort by Oldest

Add a comment...

Osman Goni Nahid ·
Full Stack Software Engineer at Inovio Dhaka

আমার ও কালকের এক্সাম এর পড়া হলো ; 🙂

Like · Reply · 👍 1 · 2 April 2014 11:47

Faisal Azam ·
Executive Officer at IFIC BANK Limited

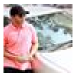
prim's algorithm টা একটু সহজ .....just code ...

Like · Reply · 👍 1 · 18 April 2014 15:58 · Edited

Mohammad Nizam Uddin ·
Software Engineer at AuthLab Limited

Obviously an awesome tutorial. Thank's Shafayet Bhaiya!!

Like · Reply · 👍 5 · 25 April 2014 09:26



Masuk Sarker Batista ·
Founder and Author at Online Earning King

prim D best 🙂

Like · Reply · 21 June 2014 13:09

Md Amir Faisal ·
Ahsanullah University of Science & Technology

Medha

Like · Reply · 👍 1 · 2 November 2015 12:10

Nilan Nilan ·
IUBAT-International University of Business Agriculture and Technology

how you are.

Like · Reply · 11 April 2016 08:28

Facebook Comments Plugin

Powered by Facebook Comments

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

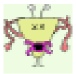
24,102 বার পড়া হয়েছে

◂ ডাটা স্ট্রাকচার: ডিসজয়েন্ট সেট(ইউনিয়ন ফাইন্ড) গ্রাফ থিওরিতে হাতেখড়ি ৭:টপোলোজিকাল সর্ট ▸

13 thoughts on “গ্রাফ থিওরিতে হাতেখড়ি ৬: মিনিমাম স্প্যানিং ট্রি(ক্রুসকাল অ্যালগোরিদম)”

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

---



Ronok1307

জুন ১৪, ২০১২ *at* ৬:১৭ *pm*

Great tutorial. But I have one problem. In your Union function you say that it’s ok to use either **par[u]=v** or **par[v]=u**. But for an ACM problem, specifically problem 544, while using Disjoint Set for Kruskal, I am getting WA with one and AC for the other. What could be the problem?

Reply

শাফায়েত

জুন ১৪, ২০১২ *at* ৬:২৩ *pm*

আমি ঠিক নিশ্চিত না সমস্যা কোথায়। যেকোনো একটা লিখলেই কাজ করার কথা। আমি এইমাত্র ৯০৮ নম্বরের কোড ট্রাই করে দেখলাম, দুটাতেই accepted হয়েছে। কোডটা দিয়ে দিলাম এখানে।

Reply

হাসান

ডিসেম্বর ২৪, ২০১২ *at* ১:২২ *am*

একই ভাবে এরপ CE তারপর DF,AB কে যুক্ত করবো:
এরপর সবথেকে ছোট এজ হলো BD, এটাকে আমরা নিতে পারবোনা কারণ BD নিলে একটি সাইকেল তৈরি হয়ে,B থেকে D তে যাবার রাস্তা আগে থেকেই আছে,তাই এজটি নেয়ার কোনো দরকার নেই।

> *AB* এর পরতো সবচেয়ে ছোট *edge* তো *BE*, তাহলে *BD* নিয়ে কাজ করবো কেনো?

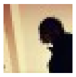
Reply

শাফায়েত

ডিসেম্বর ২৪, ২০১২ *at* ১০:৪৫ *pm*

একটু টাইপিং মিসটেক আছে, আমি ঠিক করে দিচ্ছি।

Reply

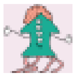


Kfoozminus

জানুয়ারি ২১, ২০১৩ *at* ২:১০ *am*

ভাইয়া একটা ব্যাপার… 29 নম্বর লাইনে pr[u]=v না লিখে pr[u]=pr[v] লিখলে find() এর কাজটা আরো কমে যেত না?

Reply

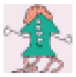


Ali

মার্চ ২৮, ২০১৪ *at* ৬:০২ *pm*

ভাইয়া c এর কোড নাই?

Reply



Ali

মার্চ ২৮, ২০১৪ *at* ৬:০৩ *pm*

ইমপ্লিমেন্টেশন টা c তে করলে বুজতে সুবিধা হত

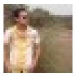
Reply

Osman Goni Nahid

এপ্রিল ৩, ২০১৪ *at* ১২:৪৯ *am*

ভাইয়া জাভা তে কোড টা হইলে আর ও ভাল হইতো ।

Reply



Rafiqul Islam Jack

জুনe ১৪, ২০১৪ *at* ৬:১৭ *pm*

Shafaet Vai, would you explain this line, if(count==n-1) break;

Reply

Pingback: গ্রাফ থিওরি এবং একটি রুপকথার গল্প | Shipu's Blog

Pingback: শাফায়েতের ব্লগ » Blog Archive

Pingback: শাফায়েতের ব্লগ » Blog Archive

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

5 × **2** =

Post Comment

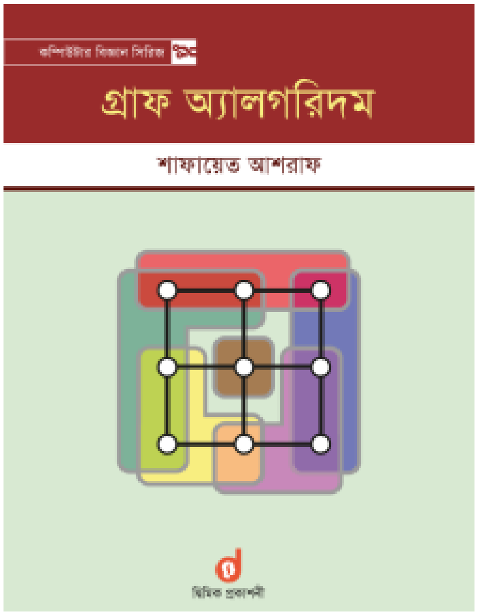
phonetic probhat english

সাবস্ক্রাইব

Powered by OneAll Social Login

আমার সম্পর্কে

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



Like Share 2.7k

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