



# শাফায়েতের ব্লগ

প্রোগ্রামিং ও অ্যালগরিদম টিউটোরিয়াল

Home

অ্যালগরিদম নিয়ে যত লেখা!

আমার সম্পর্কে...

## গ্রাফ থিওরিতে হাতেখড়ি ৫: মিনিমাম স্প্যানিং ট্রি(প্রিম অ্যালগোরিদম)

📅 আগস্ট ৪, ২০১১ by শাফায়েত



(সিরিজের অন্যান্য পোস্ট)

একটি গ্রাফ থেকে কয়েকটি নোড আর এজ নিয়ে নতুন একটি গ্রাফ তৈরি করা হলে সেটাকে বলা হয় **সাবগ্রাফ**। স্প্যানিং ট্রি হলো এমন একটি সাবগ্রাফ যেটায়:

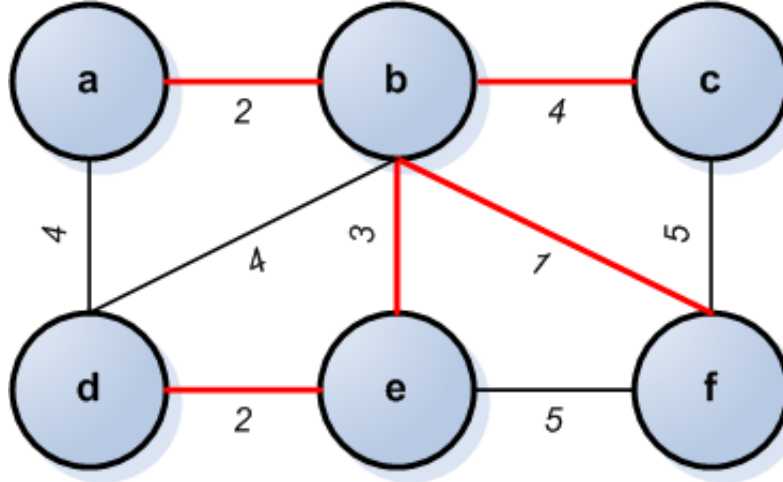
“ \* মূল গ্রাফের সবগুলো নোড আছে।

\* সাবগ্রাফটি একটি ট্রি। ট্রিতে কখনো সাইকেল থাকেনা, এজ থাকে  $n - 1$  টি যেখানে  $n$  হলো নোড সংখ্যা।

একটি গ্রাফের অনেকগুলো স্প্যানিং ট্রি থাকতে পারে, যে ট্রি এর এজ গুলোর কস্ট/ওয়েট এর যোগফল সব থেকে কম সেটাই মিনিমাম স্প্যানিং ট্রি। আমরা এই লেখায় প্রিম অ্যালগোরিদমের সাহায্যে মিনিমাম স্প্যানিং ট্রি বের করা শিখবো।

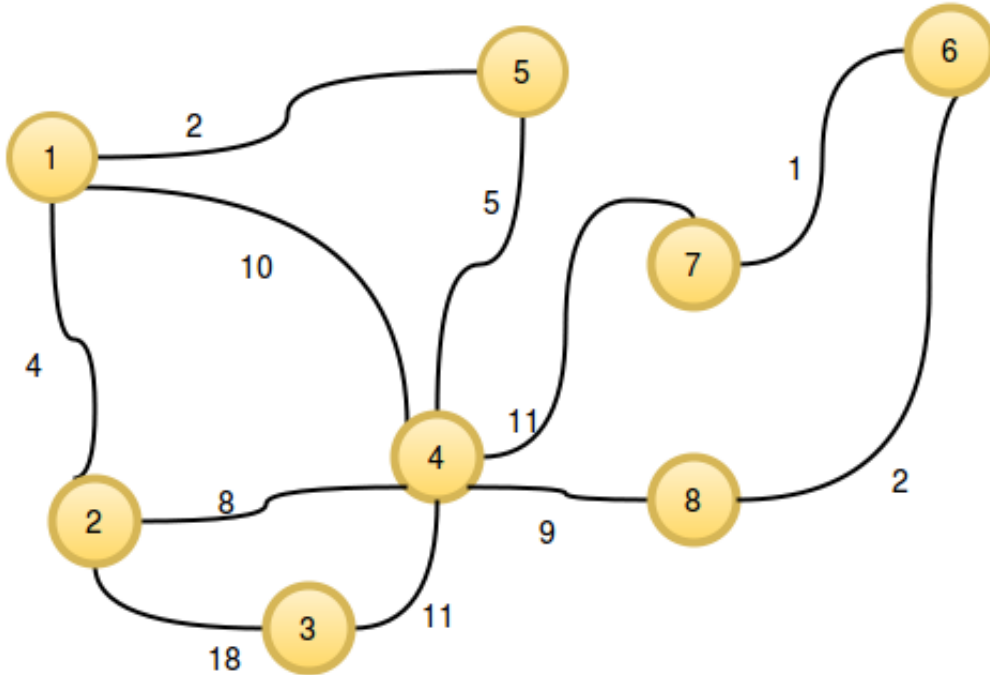
মনে করি নিচের গ্রাফের প্রতিটি নোড হলো একটি করে বাড়ি। আমাদের বাড়িগুলোর মধ্যে টেলিফোন লাইন বসাতে হবে। আমরা চাই সবথেকে কম খরচে লাইন বসাতে। এজ গুলোর ওয়েট লাইন বসানোর খরচ নির্দেশ

করে:



আমরা অনেক ভাবে লাইন বসাতে পারতাম। ছবিতে লাল এজ দিয়ে টেলিফোন লাইন বসানোর একটি উপায় দেখানো হয়েছে। টেলিফোন লাইনগুলো একটি সাবগ্রাফ তৈরি করেছে যেটায় অবশ্যই  $n - 1$  টি এজ আছে,কোনো সাইকেল নেই কারণ অতিরিক্ত এজ বসালে আমাদের খরচ বাড়বে,কোনো লাভ হবেনা। মিনিমাম স্প্যানিং ট্রি বের করার সময় আমরা এমন ভাবে এজগুলো নিবো যেন তাদের এজ এর যোগফল মিনিমাইজ হয়।

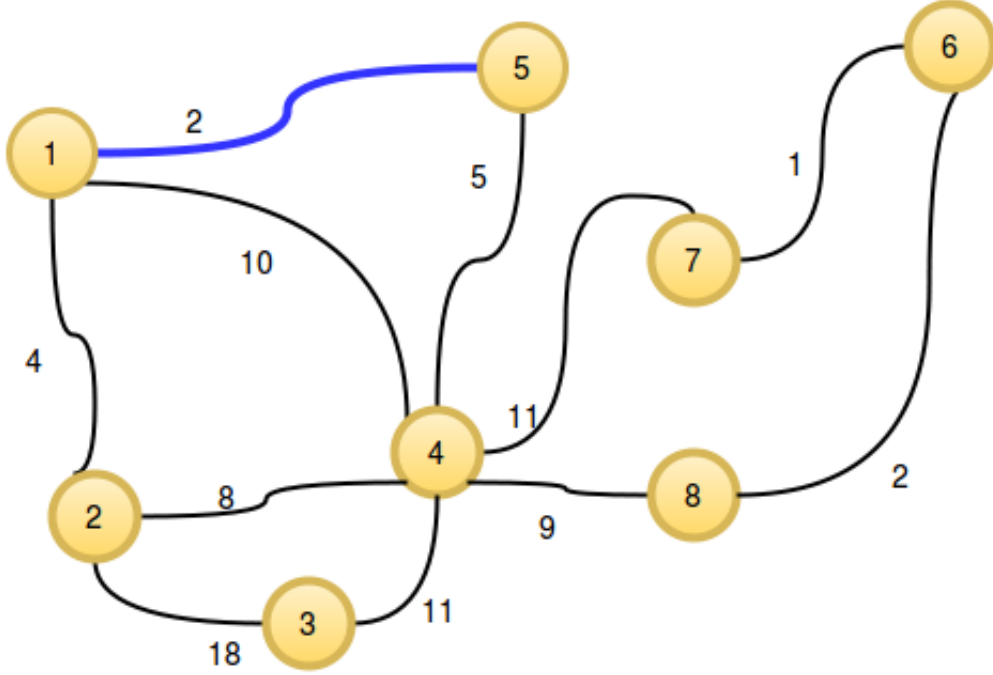
এখন নিচের গ্রাফ থেকে কিভাবে আমরা মিনিমাম স্প্যানিং ট্রি বের করব?



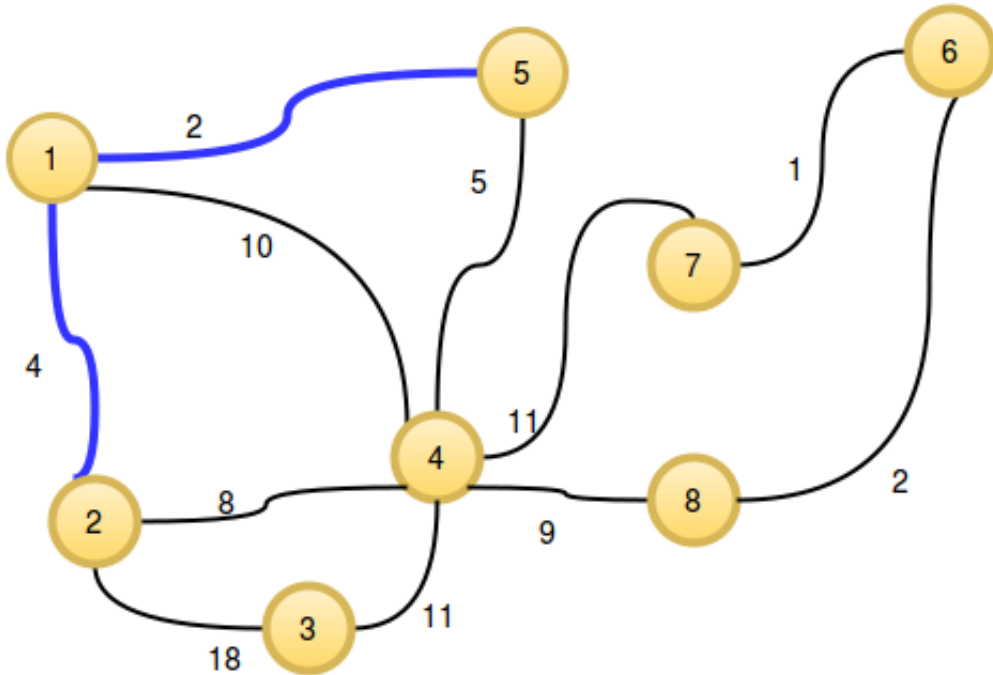
গ্রিডি(greedy) অ্যাপ্রোচে খুব সহজে মিনিমাম স্প্যানিং ট্রি বের করা যায়। আমরা এখন প্রিমস অ্যালগোরিদম কিভাবে কাজ করে দেখব। তুমি যদি আগে ক্রসকাল শিখতে চাও তাহলেও সমস্যা নেই, **সরাসরি পরের পর্বে** চলে যেতে পারো।

^  
top

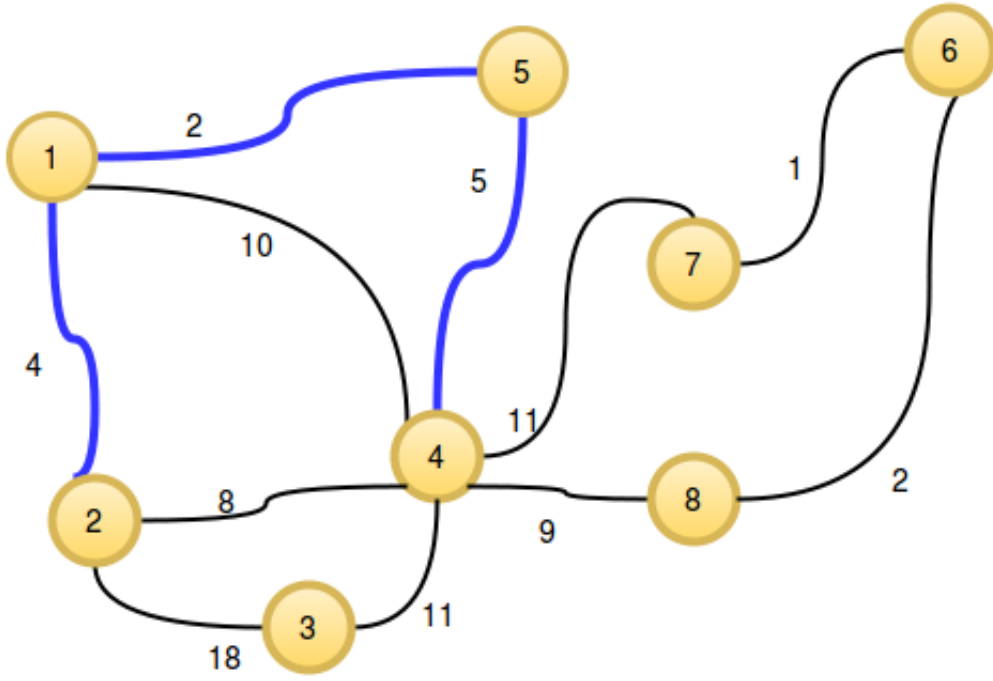
আমরা প্রথমে যেকোনো একটি সোর্স নোড নিব। ধরি সোর্স হলো ১। ১ থেকে যতগুলো এজ আছে সেগুলোর মিনিমাম টিকে আমরা সাবগ্রাফে যোগ করব। নিচের ছবিতে নীল এজ দিয়ে বুঝানো হচ্ছে এজটি সাবগ্রাফে যুক্ত করা হয়েছে:



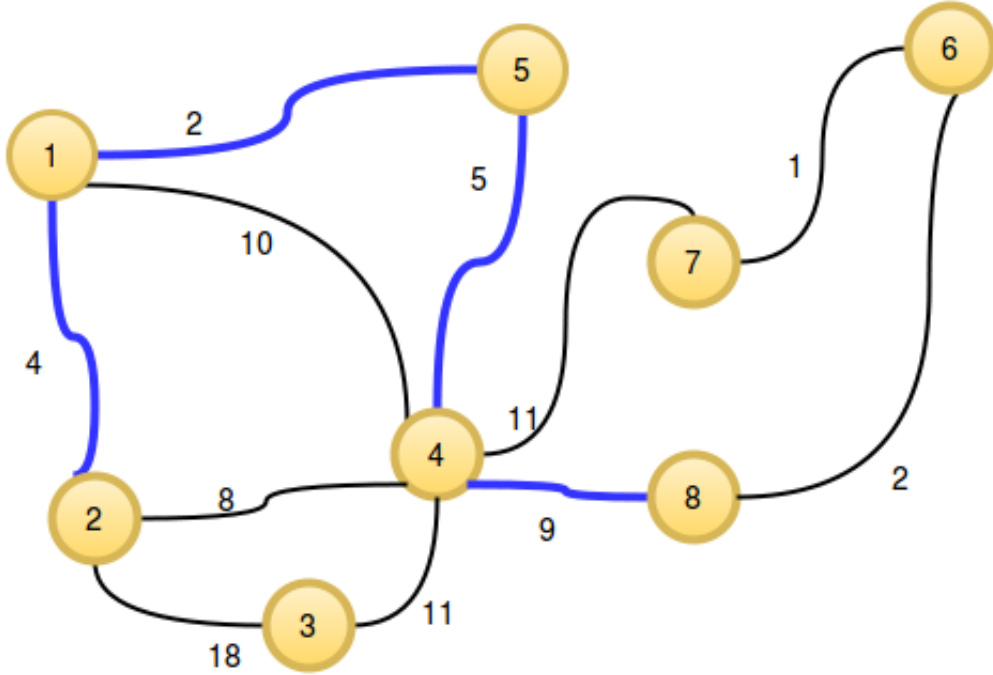
এবার সোর্স ১ এবং ৫ নম্বর নোড থেকে মোট যত এজ আছে(আগের এজগুলো সহ) তাদের মধ্যে মিনিমাম টি নিব:



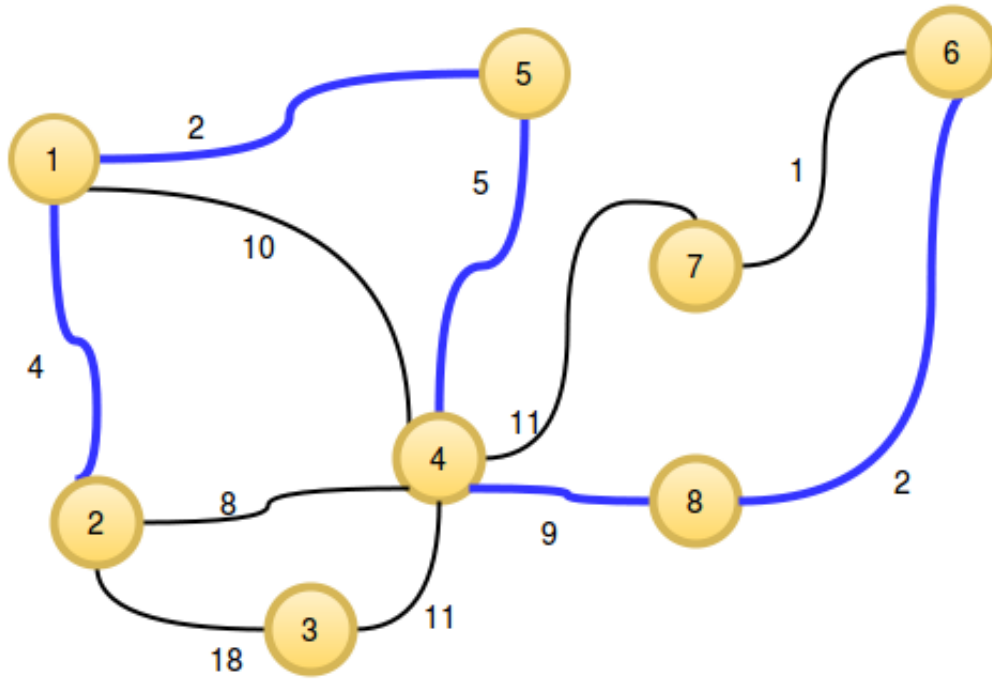
এবার নিব ১,২ এবং ৫ নম্বর নোড থেকে মোট যত এজ আছে(আগের এজগুলো সহ) তাদের মধ্যে মিনিমাম:



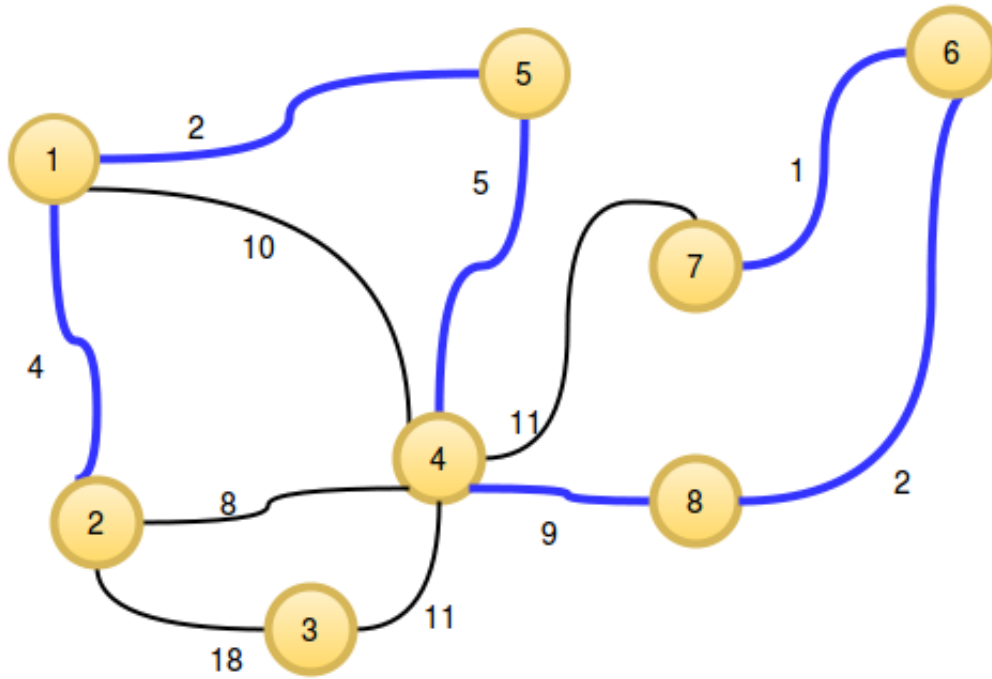
পরের ধাপটি গুরুত্বপূর্ণ। ১,২,৫,৮ থেকে যত এজ আছে তাদের মধ্য মিনিমাম হলো ২-৮, কিন্ত ২ নম্বর নোড এবং ৮ নম্বর নোড দুইটাই অলরেডি সাবগ্রাফের অংশ,তারা আগে থেকেই কানেক্টেড,এদের যোগ করলে সাবগ্রাফে সাইকেল তৈরি হবে,তাই ২-৮ এজটি নিয়ে আমাদের কোনো লাভ হবেনা। আমরা এমন প্রতিবার এজ নিব যেন নতুন আরেকটি নোড সাবগ্রাফে যুক্ত হয়। তাহলে ৮-৬ হবে আমাদের পরের চয়েস।



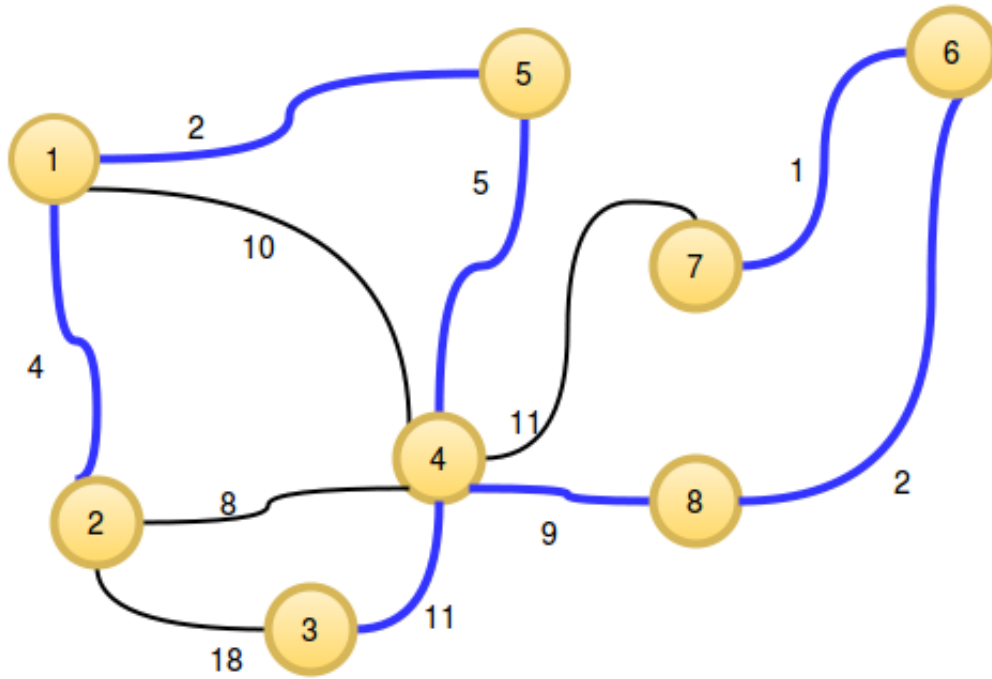
এরপর ৮-৬ যোগ করবো:



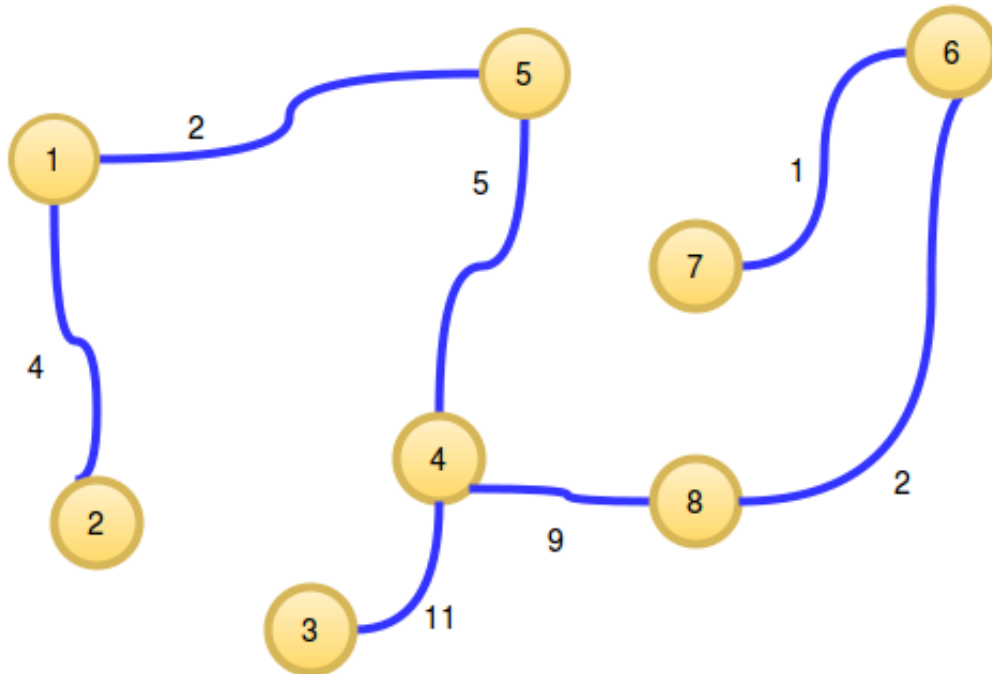
এরপর ৬-৭:



সবশেষে ৪-৩ যোগ করবো:



নীলরং এর এই সাবগ্রাফটাই আমাদের মিনিমাম স্প্যানিং ট্রি। বাকি এজগুলো মুছে দিলে থাকে:



তাহলে টেলিফোন লাইন বসাতো মোট খরচ:  $8 + 2 + 5 + 11 + 9 + 2 + 1 = 38$ । একটি গ্রাফে এক বা একাধিক মিনিমাম স্প্যানিং ট্রি থাকতে পারে।

আমাদের সুডোকোড হবে এরকম:

```

1 * Input: A non-empty connected weighted graph with vertices V and edges E (the weights can be
2 * Initialize: Vnew = {x}, where x is an arbitrary node (starting point) from V, Enew = {}
3 * Repeat until Vnew = V:
4   o Choose an edge (u, v) with minimal weight such that u is in Vnew and v is not
5   (if there are multiple edges with the same weight, any of them may be picked)
6   o Add v to Vnew, and (u, v) to Enew

```

এখন মাথায় প্রশ্ন আসতে পারে কি ভাবে প্রিমস অ্যালগোরিদম ইম্প্লিমেন্ট করব? বারবার লুপ চালিয়ে নেইভ অ্যাপ্রোচে কোড লিখলে তোমার কোড টাইম লিমিটের মধ্যে রান না করার সম্ভাবনাই বেশি।

রানটাইম কমাতে প্রায়োরিটি কিউ ব্যবহার করতে পারো। যখন নতুন একটা নোড Vnew তে যোগ করছো তখন সেই নোডের অ্যাডজেসেন্ট সবগুলো এজ প্রায়োরিটি কিউতে ঢুকিয়ে রাখতে হবে। এখন প্রায়োরিটি কিউ থেকে সবথেকে মিনিমাম ওয়েটের এজটা লগারিদম কমপ্লেক্সিটিতে খুঁজতে পারবে। মোট কমপ্লেক্সিটি হবে  $O(E \log E)$ । তবে এজের বদলে কিউতে নোড পুশ করে  $O(E \log V)$  তে কমপ্লেক্সিটি নামিয়ে আনা যায়, সেটা কিভাবে করা যায় চিন্তা করে বের করো।

অ্যালগোরিদমটা ইম্প্লিমেন্ট করার পর অবশ্যই নিচের সমস্যা গুলো সমাধানের চেষ্টা করবে।

<http://uva.onlinejudge.org/external/5/544.html>(Straight forward)

<http://uva.onlinejudge.org/external/9/908.html>

<http://uva.onlinejudge.org/external/100/10034.html>(Straight forward)

<http://uva.onlinejudge.org/external/112/11228.html>

<http://uva.onlinejudge.org/external/104/10462.html>(2nd best mst)

spoj:

<http://www.spoj.pl/problems/MST/>(Straight forward)

মিনিমাম স্প্যানিং ট্রি বের করার জন্য আরেকটি অ্যালগোরিদম আছে যা **ক্রসকাল অ্যালগোরিদম** নামে পরিচিত। পরের **পর্বে** আমরা সেটা শিখবো।

## ফেসবুকে মন্তব্য

6 comments

## 6 Comments

Sort by Oldest



Add a comment...

**Engr Mithun Dey** ·

Management Trainee IT at Ventura Lather Ware BD Manufacturing Ltd.

good

Like · Reply · 1 · 13 February 2013 08:58

**Faisal Azam** ·

Executive Officer at IFIC BANK Limited

thanks for help.....

Like · Reply · 18 April 2014 15:38

**Hafizul Islam Himel** ·

Engineer at Student

thnx sir

Like · Reply · 1 · 12 July 2016 11:05

**Anwar Hasan Shuvo** ·

Project Mentor at Daffodil International University

Very easy for making out.Thank you!

Like · Reply · 8 August 2016 08:52

**চৌধুরী আদি** ·

স্বপ্ন দেখি,দেখাই at দেশ তো বদলাতে হবে আমাকেই

গ্রাফ বুঝানোর সময় ৬ থেকে ৭ এ যাওয়া হয়েছে, কিন্ত সেটা না  
লিখে কস্ট টা লিখে দেয়া হয়েছে ৬-১ 😊 এটা বোধহয় " এরপর 6 -  
7" হবে ০ঃ০

Like · Reply · 1 · 7 September 2016 11:37

**মেহেদী** ·

Patuakhali Science and Technology University

helped me much...But,implementation code ta dile aro valo hoto.

Like · Reply · 8 September 2016 03:24

Facebook Comments Plugin

Powered by Facebook Comments





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

34,837 বার পড়া হয়েছে

◀ প্রাইম জেনারেটর (Sieve of Eratosthenes)

ডাটা স্ট্রাকচার: ডিসজয়েন্ট সেট(ইউনিয়ন ফাইন্ড) ▶

22 thoughts on "গ্রাফ থিওরিতে হাতেখড়ি ৫: মিনিমাম স্প্যানিং ট্রি(প্রিম অ্যালগোরিদম)"

Pingback: গ্রাফ থিওরিতে হাতেখড়ি ৬: মিনিমাম স্প্যানিং ট্রি | ACMSolver - Bangla



মুন্না

ডিসেম্বর ৮, ২০১১ at ২:১৩ am

"priority queue ডিক্লেয়ার করে সেটায় 'data' টাইপের ভ্যারিয়েবল পুশ করলে Q তে অটোমেটিক সর্ট হয়ে যাবে।" ...ভাইয়া এখানে একটু সমস্যা হচ্ছে, একটু যদি ডিটেইলস লিখতেন যে কিভাবে data টাইপের structure দিয়ে minimum priority queue তৈরি করতে হয়, খুব উপকার হত। আর আপনার পুরো কোড টা একটু দেখতে চাই...প্লিজ।

আপনাকে অভিবাদন এত সুন্দর একটা টিউটোরিয়াল লেখার জন্য।

Reply



শাফায়েত

ডিসেম্বর ৮, ২০১১ at ৮:৫৭ pm

কোডে সামান্য ভুল আছে, return cost < p.cost; এর জায়গায় return cost > p.cost; হবে।

আমরা ইন্টিজার, ডাবলে সহজেই +, < ইত্যাদি অপারেটর ব্যবহার করতে পারি। কিন্তু স্ট্রাকচারগুলো কাস্টম ডাটাটাইপ, এগুলোর জন্য অপারেটর নিজে ডিফাইন করে দিতে হয়। struct data { int u,v,cost;}; ডাটা টাইপের দুটি ভ্যারিয়েবল তুমি তুলনা করবে কিভাবে? u,v বা cost এর ভিত্তিতে বা এ সবগুলোর ভিত্তিতে ছোট-বড় তুলনা করা যেতে পারে তবে তোমাকে সেটা ডিফাইন করে দিতে হবে। উপরের কোডে ঠিক এভাবেই ডিফাইন করা হয়েছে ওভারলোডিং এর মাধ্যমে, এ ক্ষেত্রে cost অনুযায়ী তুলনা করা হয়েছে।

প্রায়োরিটি কিউ যখন তোমার ডাটা গুলোকে সর্ট করবে তখন সে "<" ব্যবহার করে তুলনা করবে,তাই আমরা "<" এর আচরণ ডিফাইন করে দিয়েছি। অনেক প্রোগ্রামার ওভারলোডিং এর বিরুদ্ধে কারণ কোড readability হারায়,এ কারণে জাভাসহ অনেক হাইলেভেল ল্যাংগুয়েজ এটি সাপোর্ট করেনা। উইকিতে একটা ছোট আর্টিকেল আছে পুরোটা পড়ে ফেলো:

[http://en.wikipedia.org/wiki/Operator\\_overloading](http://en.wikipedia.org/wiki/Operator_overloading)

Reply



মুন্না

ডিসেম্বর ১০, ২০১১ at ১১:৪৪ pm

ধন্যবাদ উত্তর দেয়ার জন্য । আরেকটা সমস্যায় পরেছি তা হচ্ছে...

“পরের ধাপটি গুরুত্বপূর্ণ। ১,২,৫,৮ থেকে যত এজ আছে তাদের মধ্য মিনিমাম হলো ২-৮, কিনুত ২ নম্বর নোড এবং ৮ নম্বর নোড দুইটাই অলরেডি সাবগ্রাফের অংশ,তারা আগে থেকেই কানেক্টেড,এদের যোগ করলে সাবগ্রাফে সাইকেল তৈরি হবে,তাই ২-৮ এজটি নিয়ে আমাদের কোনো লাভ হবেনা। আমরা এমন প্রতিবার এজ নিব যেন নতুন আরেকটি নোড সাবগ্রাফে যুক্ত হয়।”

আমি যে কোড টা করেছি তাতে সাইকেল তৈরী হয়ে যাচ্ছে 😞 কোন উপায় খুজে পাচ্ছিনা... একটু হেল্প চাই...প্লিজ। আর আপনার প্রিম'স অ্যালগরিদম এর কোড টা একটু দেখতে চাই,আমাকে মেইল করে দিলে খুব ভাল হয়।

অসংখ্য ধন্যবাদ সময় দেয়ার জন্য।

Reply



শাফায়েত

ডিসেম্বর ১৪, ২০১১ at ১:০৯ am

আমি প্রিম মাত্র একবার ইম্প্লিমেন্ট করেছি,এই টিউটোরিয়ালটা লেখার সময়,কোডটা এখন নেই। তুমি যখনই কোনো এজ নিচ্ছে তখন দুপাশের নোডদুটিকে ভিজিটেড করে দাও,যখন নতুন এজ কিউ থেকে নিবে তখন আগে চেক করবে যে এজের দুপাশের দুটি নোডই ভিজিটেড নাকি,যদি দুটিই ভিজিটেড হয় তাহলে সেই এজ নেবার কোনো দরকার নেই। না পারলে তোমার কোডটি পোস্ট করতে পারো,দেখার চেষ্টা করবো।

Reply

---

Pingback: [প্রোগ্রামার শাফায়েতের যত গ্রাফ থিওরীর টিউটোরিয়াল | ACM Solver - Bangla](#)

---



**Ahbab**

নভেম্বর ২৫, ২০১২ at ১২:৪৫ am

bool operator p.cost;

ভাইয়া, এই লাইন ২ টার কাজ বুজতে পারতেছি না। প্লিজ একটু ব্যাখ্যা করে দেন।

Reply



**শাফায়েত**

নভেম্বর ২৭, ২০১২ at ১০:২৩ pm

এটাকে অপারেটর ওভারলোডিং বলে, উপরের একটা কमेंটে ব্যাখ্যা করেছি দেখো।

Reply



**Ahbab**

নভেম্বর ২৫, ২০১২ at ১২:৫২ am

bool operator p.cost;

এই ২ টা লাইনে

Reply

---

Pingback: [গ্রাফ থিওরি এবং একটি রূপকথার গল্প | codingaquarium](#)

---



**ইমতিয়াজ হাসান**

ডিসেম্বর ২৭, ২০১২ at ৯:৩১ pm

আগের কমেন্টটা ডিলিট করে দিবেন প্লিজ।

ভাইয়া priority queue নিয়েই যন্তো সমস্যা।

```
1 struct data
2 { int u,v,cost;
3     bool operator < ( const data& p ) const { //overloading operator
4         return cost > p.cost;
5     }
6 };
```

নরমাল সর্ট করার সময় যেভাবে compare ফাংশান দিয়ে কাজ করি সেটা কীভাবে করবো? ওভারলোডিং করতে আমার সমস্যা হয় কীনা!

```
1 struct data{
2     int a,b,c;
3 };
4 data arr[size];
5 bool comp(data a,data b){
6     return a.c>b.c;
7 }
8 sort(arr,arr+size,comp);
```

আর ভেরিয়াবল ডিক্লেয়ার করবো কীভাবে?

```
1 priority_queue < data > pq;
```

compare function এর কী ব্যবস্থা করবো?

```
1 priority_queue < data ,comp > pq; ???
```

আর পুশ কি এইভাবে করতে হবে ভেক্টরের নিয়মে?

```
1 struct TT{
2     int a,b;
3 };
4 vector v;
5 for(i=0;i<5;i++){
6     TT t;
7     t.a=10*i;
8     t.b=10+i;
9     v.push_back(t);
10 }
```

Reply



শাফায়েত

ডিসেম্বর ২৭, ২০১২ at ১১:০৬ pm

তুমি এই [লিংকটা দেখো](#), আমার মনে হয় এটা দেখার পরে তোমার সমস্যা মিটে যাবে, এরপরেও সমস্যা থাকলে জানাও।

Reply



**ইমতিয়াজ হাসান**

ডিসেম্বর ২৭, ২০১২ at ১১:৪৫ pm

ধন্যবাদ ভাইয়া।

Reply



**অনিন্দ্য সুন্দর পাল**

আগস্ট ৪, ২০১৩ at ৫:৩৫ pm

"যেহেতু এটা অ্যালগোরিদমের টিউটোরিয়াল, সি++ এর না, তাই এসব নিয়ে আর বেশি লিখবোনা, কোথায় সমস্যা হলে মন্তব্য অংশে বা আমার সাথে যোগাযোগ করে জানাতে পারো, আর তার আগে [এই লিংকটা একবার দেখো।](#)"

ভাইয়া, এই লিংকটা কাজ করছে না।

Reply



**শাফায়েত**

আগস্ট ৪, ২০১৩ at ৬:২১ pm

লিংকটা সরিয়ে ফেলা হয়েছে। অন্য একটা লিংক বসিয়ে দিলাম।

Reply



**মুহিম মুক্তাদির**

অক্টোবর ২৮, ২০১৩ at ৪:১১ pm

ভাইয়া, আমি ইউভিএ এর ৫৪৪ নাম্বার প্রব্লেম টা implement করেছি কিনুত verdict- TLE . নিচে আমার কোড, দয়া করে সাহায্য করলে উপকৃত হতাম ০ঃ

<http://ideone.com/JRDg5A>

Reply



জিনি

মার্চ ১৯, ২০১৪ at ৬:৫৮ am

ভাইয়া এইটা দেখেন। উপরের কमेंটটা মুছে ফেইলেন, প্লিজ। 😊

```
1 #include
2 #include
3 #include
4 #include
5 #include
6
7 using namespace std;
8 struct data
9 {
10     int v,w;
11     bool operator p.w;
12 }
13 };
14 vectorvec[2000];
15 vectorcost[2000];
16 void mst(int src,int n)
17 {
18     priority_queuepq;
19     int visited[2000]={0},i,u,v,sum=0,j,p;
20     for(i=1;i<n;i++)
21     {
22         u=src;
23         visited[src]=1;
24         for(j=0;j<vec[u].size();j++)
25         {
26             v=vec[u][j];
27             if(visited[v]==0)
28             {
29                 data D; D.v=v;
30                 D.w=cost[u][j];
31                 pq.push(D);
32             }
33         }
34         do{
35             data T=pq.top();
36             src=T.v;
37             p=T.w;
38             pq.pop();}while(visited[src]==1);
39         sum+=p;
40     }
41     printf("%d",sum);
42 }
43 int main()
44 {
45     freopen("input.txt","r",stdin);
```

```
46     int n,e,n1,n2,w,i;
47     scanf("%d%d",&n,&e);
48     for(i=0;i<e;i++)
49     {
50         scanf("%d%d%d",&n1,&n2,&w);
51         vec[n1].push_back(n2);
52         vec[n2].push_back(n1);
53         cost[n1].push_back(w);
54         cost[n2].push_back(w);
55     }
56     mst(1,n);
57     return 0;
58 }
```

Reply



শাফায়েত

মার্চ ২৫, ২০১৪ at ২:৩৩ pm

ঠিকই আছে, বেশি জটিল হয়নি :)। [আমি অবশ্য কোডটা এক্সিকিউট করে দেখিনি, লজিক ঠিকই আছে মনে হচ্ছে]

Reply

Pingback: [গ্রাফ থিওরি এবং একটি রূপকথার গল্প | Shipu's Blog](#)

Pingback: [Prims Algorithm Implementation | saadnoorsalehin](#)

Pingback: [Minimum Spanning tree | hasinruetblog](#)

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

8 + 5 =



Post Comment

phonetic

probhat

english

## সাবস্কাইব

---

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 (ক্রসকাল 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

