

## শাফায়েতের ব্লগ

প্রোগ্রামিং, অ্যালগরিদম, ব্যাকএন্ড ইঞ্জিনিয়ারিং

Home

অ্যালগরিদম নিয়ে যত লেখা!

আমার সম্পর্কে...

## গ্রাফ থিওরিতে হাতেখড়ি ১২ – ম্যাক্সিমাম ফ্লো (১)

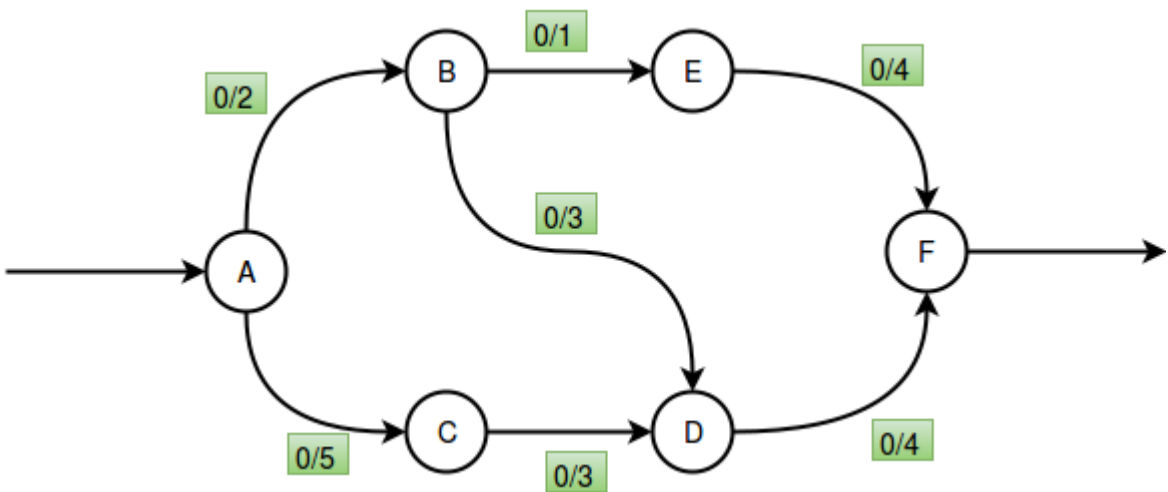
📅 জানুয়ারি ২৩, ২০১৫ by Shafaet Ashraf






এই লেখায় আমরা গ্রাফে ম্যাক্সিমাম ফ্লো বের করার অ্যালগোরিদম শিখবো। ম্যাক্স ফ্লো এর ধারণাটা ব্যবহার করে বেশ কিছু ইন্টারেস্টিং প্রবলেম সলভ করা যায়, তাই এটা শেখা খুবই গুরুত্বপূর্ণ। এই লেখাটা পড়ার আগে তোমাকে **গ্রাফ থিওরি**র বেসিক অ্যালগোরিদমগুলো, বিশেষ করে শর্টেস্ট পাথ বের করার অ্যালগোরিদমগুলো ভালো করে শিখে নিবে।

প্রথমেই আমরা দেখি একটি খুবই সাধারণ ম্যাক্স ফ্লো প্রবলেম:



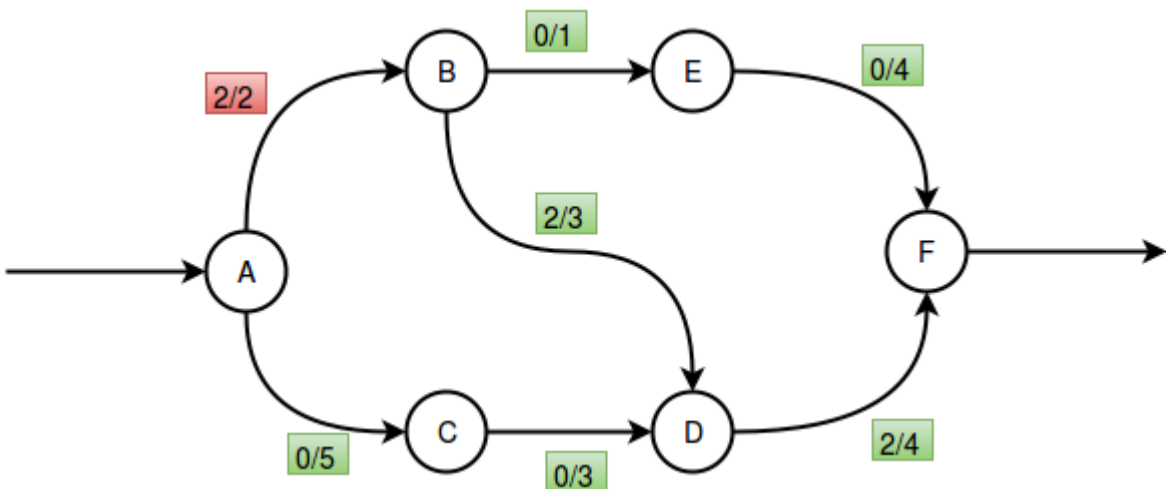
চিত্র: ১

তোমাকে চিত্র-১ এর মত একটা গ্রাফ দেয়া আছে। মনে কর গ্রাফের প্রতিটা এজ একটা করে পানির পাইপ। প্রতিটা পাইপ দিয়ে প্রতি সেকেন্ডে কত লিটার পানি প্রবাহিত হতে পারবে সেটার একটা সীমা আছে যেটাকে বলা হয় পাইপের **ক্যাপাসিটি**। আর কোনো পাইপ দিয়ে সেকেন্ডে যতটুকু পানি যাচ্ছে সেটা হলো পাইপটার **ফ্লো**। প্রতিটা এজের সাথে “ফ্লো/ক্যাপাসিটি” উল্লেখ করে দেয়া হয়েছে। যেমন A->B এজ দিয়ে সেকেন্ডে সর্বোচ্চ ২লিটার পানি যেতে পারে এবং এই মুহূর্তে সেকেন্ডে পানি যাচ্ছে ০ লিটার। শুধুমাত্র A নোড দিয়ে পানি প্রবেশ করানো যায় এবং শুধুমাত্র F নোড দিয়ে পানি বের হয়ে যায়। A কে বলা হয় **সোর্স** এবং F হলো **সিংক**। তোমাকে বলতে হবে A থেকে F এ প্রতি সেকেন্ডে সর্বোচ্চ কত লিটার পানি প্রবাহিত করা যাবে? অর্থাৎ পানির “ফ্লো” সর্বোচ্চ কত হতে পারে?

মূল সমস্যা সমাধানের আগে আমরা ছোট একটা সমস্যা সমাধান করি। A->C->D->F এই পথটা ব্যবহার করে পানি সোর্স থেকে সিংক এ গেলে সর্বোচ্চ কত লিটার পানি প্রবাহিত হতে পারবে? ছবিতে দেখা যাচ্ছে A->C এজের ক্যাপাসিটি ৫, C->D এজের ক্যাপাসিটি ৩, D->F এজের ক্যাপাসিটি ৪। এখানে সর্বনিম্ন ক্যাপাসিটি হলো ৩, তাই আমরা কোনো সময় ৩লিটারের বেশি পানি এই পথে পাঠাতে পারবোনা।

কোনো পথের সর্বনিম্ন ক্যাপাসিটির এজ সেই পথের ফ্লো নিয়ন্ত্রণ করে। বোতলের সরু মুখের সাথে তুলনা দিয়ে এই ব্যাপারটাকে বলা হয় বোটলনেক(bottleneck)। যেমন ধরো তোমার বাসায় ৫MBps এর ইন্টারনেট কানেকশন আছে, এখন তুমি একটা মুভি ডাউনলোড করতে চাচ্ছ। এখন মুভির সার্ভার থেকে তোমার বাসায় ডাটা প্যাকেট আসার আগে অনেকগুলো নেটওয়ার্ক পার হয়ে আসে, কোনো একটা নেটওয়ার্কে গতি মাত্র 1MBps। এখন নেটওয়ার্কের বাকি অংশের গতি যতই বেশি হোক না কেন তুমি 1MBps এর বেশি গতিতে ডাউনলোড করতে পারবে না, নেটওয়ার্কের সবথেকে ধীরগতির অংশই তোমার ডাউনলোডের গতি নিয়ন্ত্রণ করবে।

এখন দেখি কিভাবে সর্বোচ্চ পানির ফ্লো পাবার সমস্যাটার সমাধান করা যায়। সমাধান খুবই সহজ, আমরা প্রতিবার যেকোনো একটা করে পথ দিয়ে পানি পাঠাতে থাকবো যতক্ষণ পাইপে ক্যাপাসিটি থাকে। কিন্তু এভাবে কি আমরা সর্বোচ্চ ফ্লো বা প্রবাহ পাবো? পাবো তবে সেজন্য একটু বুদ্ধি খাটাতে হবে। প্রথমে আমরা একটা পথে পানির ফ্লো পাঠিয়ে দেখি কি ঘটে। যেকোনো একটা পথ বেছে নিতে পারি, আমি বুঝানোর সুবিধার জন্য A->B->D->F পথটা বেছে নিলাম। এই পথে সর্বনিম্ন ক্যাপাসিটি হলো ২। তাহলে এই পথে ২লিটার পানির ফ্লো পাঠিয়ে দেবার পর গ্রাফটা দেখতে হবে এরকম:

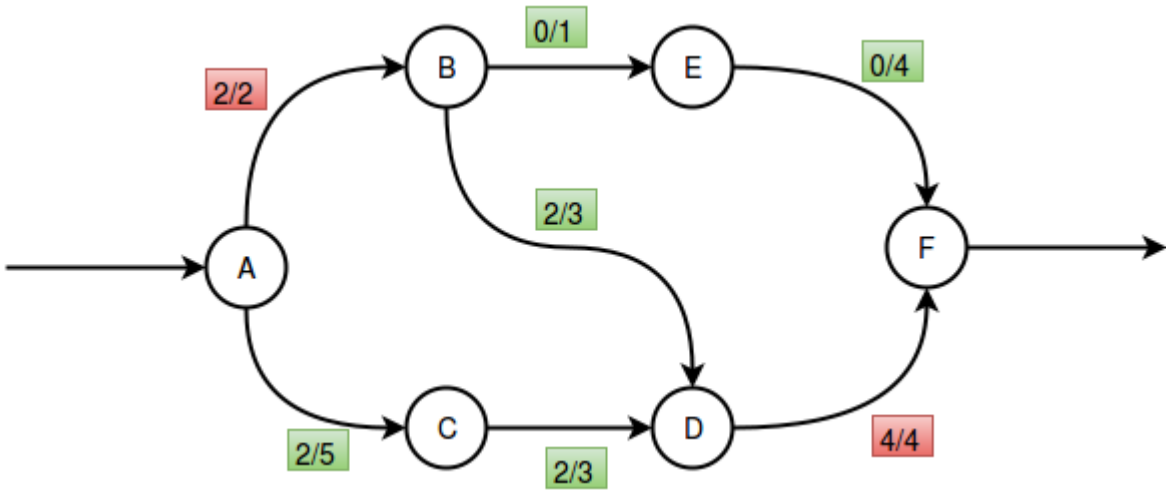


চিত্র-২:  $A \rightarrow B \rightarrow D \rightarrow F$  পথে ফ্লো পাঠানোর পর

আমরা  $A \rightarrow B \rightarrow D \rightarrow F$  পথের প্রতিটি এজের ফ্লো ২ বাড়িয়ে দিয়েছি। লক্ষ্য করো  $A \rightarrow B$  এজের আসল ক্যাপাসিটি ২ এবং ফ্লো পাঠানো হয়েছে ২, তাই এই এজ নিয়ে আর কোনো ফ্লো পাঠানো যাবে না। তেমনি  $B \rightarrow D$  এজের ক্যাপাসিটি ৩ কিন্তু ফ্লো পাঠানো হয়েছে ২, তাই এই এজটা দিয়ে আরো  $3-2=1$  ফ্লো পাঠানো সম্ভব। আমরা বলতে পারি  $B \rightarrow D$  এজের **residual ক্যাপাসিটি**=১, residual শব্দটার অর্থ হলো কোনো কিছু ব্যবহার করে ফেলার পর বেচে যাওয়া অংশ।

“ residual ক্যাপাসিটি = এজ এর ক্যাপাসিটি – ব্যবহৃত ক্যাপাসিটি বা ফ্লো এর পরিমাণ।

এবার একইভাবে  $A \rightarrow C \rightarrow D \rightarrow F$  পথটা নির্বাচিত করি। এই পথে  $D \rightarrow F$  এজের আসল ক্যাপাসিটি ৪ হলেও ২ ফ্লো আগেই পাঠানো হয়েছে, তাই বর্তমান ক্যাপাসিটি বা **residual ক্যাপাসিটি**  $4-2=2$ । এই পথটা দিয়ে ২ ফ্লো পাঠানো সম্ভব।

চিত্র-৩:  $A \rightarrow C \rightarrow D \rightarrow F$  পথে ফ্লো পাঠানোর পর

আমরা এখন মোট  $2+2=4$  ফ্লো পেলাম, অর্থাৎ পানি প্রবাহের হার এখন সেকেন্ডে ৪ লিটার। এই গ্রাফে কি আরো বেশি পানির ফ্লো পাঠানো সম্ভব?  $A \rightarrow B$  এবং  $D \rightarrow F$  এজ এরই মধ্যে সর্বোচ্চ ক্যাপাসিটিতে পৌঁছে গেছে, এই ২টা এজের কোনোটাকে না নিয়ে সিংক এ যাবার পথ গ্রাফে নেই। তাই দেখে মনে হতে পারে আর ফ্লো পাঠানো সম্ভব না। কিন্তু আমরা এখন বুদ্ধি খাটিয়ে আরো বেশি ফ্লো পাঠিয়ে দিব!

আমাদের এখন প্রতিটা এজের residual ক্যাপাসিটি ব্যবহার করে **residual গ্রাফ** আঁকতে হবে কিভাবে আরো ফ্লো পাঠাবো সেটা বুঝতে হবে। residual গ্রাফ আকার নিয়ম হলো:

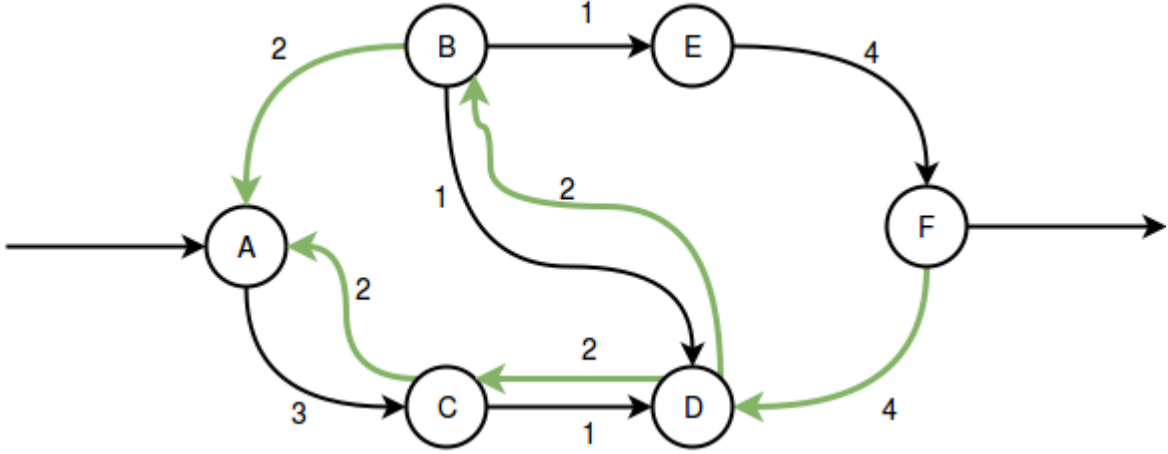
“ ১. গ্রাফের প্রতিটা এজ  $(u,v)$  এর ক্যাপাসিটি হবে এজ টার residual ক্যাপাসিটির সমান।

২. প্রতি এজ  $(u,v)$  এর জন্য উল্টা এজ  $(v,u)$  এর residual ক্যাপাসিটি হবে  $(u,v)$

এজ এ ফ্লো এর সমান।

তারমানে কোনো এজ দিয়ে যতটুকু ফ্লো পাঠিয়েছি সেটা হবে উল্টো এজের residual ক্যাপাসিটি। তাহলে residual গ্রাফে মূল এজ এবং উল্টো এজের residual ক্যাপাসিটির যোগফল হবে মূল এজ এজের মোট ক্যাপাসিটির সমান।

চিত্র-৪ এর গ্রাফটার residual গ্রাফটা হবে এরকম:



চিত্র-৪: ৩নং চিত্রের গ্রাফের residual গ্রাফ

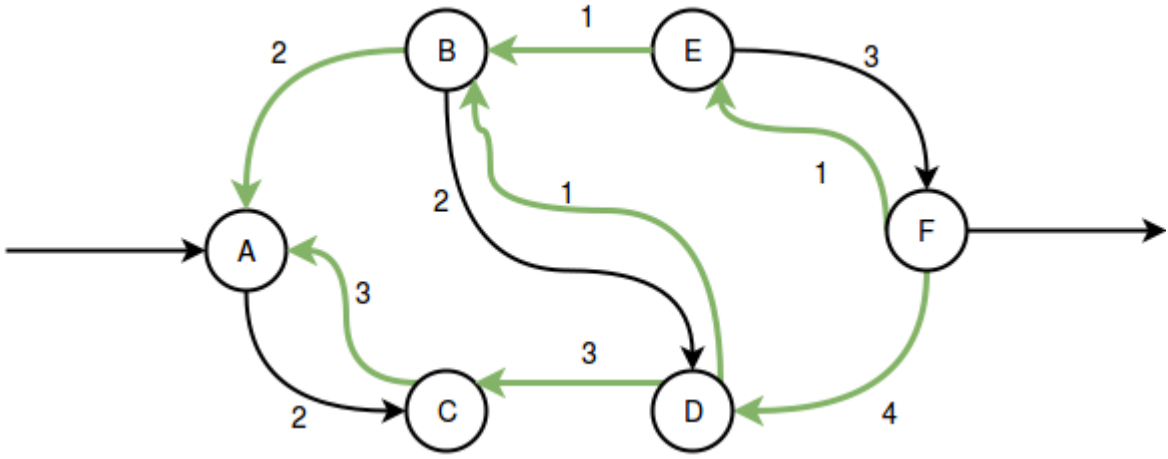
সবুজ রঙ দিয়ে উল্টো এজ এবং তাদের residual ক্যাপাসিটি দেখানো হয়েছে। o ক্যাপাসিটির এজগুলোকে গ্রাফে একে দেখাইনি। A->C এজ দিয়ে আগে ২ ফ্লো পাঠানো হয়েছে বলে উল্টো এজ এর residual ক্যাপাসিটি ২, এবং এজটার আরো ৩ ফ্লো পাঠানোর ক্ষমতা আছে তাই মূল এজ এর residual ক্যাপাসিটি ৩। ঠিক এভাবে অন্যান্য এজগুলো আঁকা হয়েছে, তুমি নিজে একে যাচাই করে নিতে পারো ঠিকমত বুঝেছো নাকি বা আমার ছবিতে ভুল আছে নাকি।

এখন উল্টো এজ এ ফ্লো পাঠানোর মানে কি? আমরাতো পাইপের উল্টো দিক দিয়ে পানি পাঠাতে পারবো না বা রাস্তার উল্টো দিক দিয়ে গাড়ি চালাতে পারবো না। উল্টো দিকে ফ্লো পাঠানোর মানে হলো মূল ফ্লো টাকে বাতিল করে দেয়া! আমরা যদি D->B এজ দিয়ে ২ ফ্লো পাঠাই তার মানে B->D এজ দিয়ে আসা ২লিটার পানির প্রবাহকে আমরা বাতিল করে দিচ্ছি। তখন residual গ্রাফে মূল এজের residual ক্যাপাসিটি যাবে বেড়ে, এবং উল্টো এজের residual ক্যাপাসিটি যাবে কমে, যোগফল আগের মতই থাকবে।

এখন চিন্তা করে দেখো চিত্র-৪ দিয়ে আরো ফ্লো সিংক এ পাঠানো যায় নাকি। মনে রাখবে চিত্র-৩ আর চিত্র-৪ এ একই গ্রাফকেই ভিন্নভাবে আঁকা হয়েছে।

চিত্র ৪ এ দেখা যাচ্ছে A->C->D->B->E->F পথে আরো ১ ফ্লো পাঠানো সম্ভব! এই পথে D->B হলো উল্টো এজ।

তারমানে D->B এজ এর ১ ফ্লো বাতিল করে দিয়ে আমরা গ্রাফে মোট ফ্লো ১বাড়িয়ে ফেলতে পারছি। এখন residual গ্রাফ কিরকম হবে দেখি:



চিত্র-৫: A->C->D->B->E->F পথে ফ্লো পাঠানোর পর residual গ্রাফ

তুমি যদি এ পর্যন্ত বুঝে থাকো তাহলে চিত্র-৫ থেকে মূল গ্রাফটাও নিজে একে নিতে পারবে। চিত্র-৪ এ C->D এজ দিয়ে ১ ফ্লো পাঠানো হয়েছে। তাই চিত্র-৫ এ C->D এজের residual ক্যাপাসিটি কমে  $১-১=০$  হয়ে গিয়েছে এবং D->C এজের residual ক্যাপাসিটি বেড়ে  $২+১=৩$  হয়ে গিয়েছে। এভাবে সবগুলো এজ আঁকা হয়েছে।

এই গ্রাফে সোর্স থেকে সিংক এ পানি পাঠানোর আর কোনো পথ নেই। আমরা ৩টি পথে মোট  $২+২+১=৫$  ফ্লো পেয়েছি, অর্থাৎ পানি প্রবাহের হার প্রতি সেকেন্ডে ৫ লিটার। এটাই গ্রাফটার জন্য ম্যাক্সিমাম ফ্লো।

residual গ্রাফে আমরা যখন একটা পথ বের করি সেই পথের একটা বিশেষ নাম আছে, সেটা হলো **অগমেন্টেড পাথ(Augmented path)**। তাহলে আমাদের অ্যালগোরিদম খুব সহজ, যতক্ষণ সম্ভব আমরা residual graph এ একটা অগমেন্টেড পাথ খুঁজে বের করবো এবং সেই পথে ফ্লো পাঠিয়ে দিবো! এটার নাম Ford-Fulkerson অ্যালগোরিদম। এটার সুডোকোড এরকম:

```

1 Input: A graph  $G = (V, E)$  with flow capacity cap, source node s and sink node t.
2 Output: Calculate maximum flow from s to t.
3 Initialize:
4     1. total_flow=0
5     2. Residual Capacity  $C_f(u, v) = \text{cap}(u, v)$  for each edge  $(u, v)$  in the graph
6 Algorithm:
7     1. While there is a path from s to t such that  $C_f[u][v] > 0$  for all edges  $(u, v)$  in path:
8         1.1. min_res_cap = minimum residual capacity among all the edges  $(u, v)$  in the path.
9         1.2. For each edge  $(u, v)$  in the path:
10             $C_f(u, v) = C_f(u, v) - \text{min\_res\_cap}$ 
11             $C_f(v, u) = C_f(v, u) + \text{min\_res\_cap}$ 
12         1.3. total_flow = total_flow + min_res_cap
13     3. Return total_flow

```

সহজ কথায় একটা ২-ডি অ্যারে  $C_f[u][v]$  তে প্রতিটা এজের residual ক্যাপাসিটি থাকবে, যে এজগুলো গ্রাফে নাই সে এজের residual ক্যাপাসিটি ০। এখন তুমি সোর্স থেকে সিংক এ যাবার এমন একটা পথ বের করবে যে পথের প্রতিটা এজের residual ক্যাপাসিটি ০ থেকে বড়। পথের এজগুলোর মধ্যে মিনিমাম residual ক্যাপাসিটির মান খুঁজে বের করবে। ফ্লো পাঠানোর পর প্রতিটা এজের residual ক্যাপাসিটি কমে যাবে এবং উল্টা এজের residual ক্যাপাসিটি

বেড়ে যাবে। মিনিমাম residual ক্যাপাসিটিটা মোট ফ্লো এর মানের সাথে যোগ হতে থাকবে। সবশেষে চাইলে তুমি আসল ক্যাপাসিটি থেকে residual capacity বিয়োগ করে কোনো এজ এ ফ্লো এর পরিমাণ নির্ণয় করতে পারো।

গ্রাফটা ডিরেক্টেড হতে হবে এমন কোনো কথা নেই। বাইডিরেকশনাল এজও থাকতে পারে। A-B বাইডিরেকশনাল এজের ক্যাপাসিটি ১০ হলে  $C_f[A][B] = C_f[B][A] = 10$  হবে। এখন আগের মতই অগমেন্টেড পাথ বের করে ম্যাক্সিমাম ফ্লো বের করতে পারবে। তবে এক্ষেত্রে কোন নোডে ফ্লো কত হচ্ছে সেটা বের করতে হলে তোমাকে আরেকটু বুদ্ধি খাটাতে হবে।

অ্যালগোরিদমে ২নম্বর ধাপে পাথ বের করার জন্য **বিএফএস/ডিএফএস/বেলম্যান ফোর্ড** ইত্যাদি অ্যালগোরিদম ব্যবহার করা যেতে পারে। প্রবলেমের ধরণ অনুযায়ী এই ধাপে একেক অ্যালগোরিদম একেক ধরণের সুবিধা দিবে, সেগুলো তুমি প্রবলেম সলভ করতে করতে জানতে পারবে। আর পাথ বের করার জন্য বিএফএস বা যেটাই ব্যবহার করোনা কেন, প্রতিটা নোডে যাবার সময় নোডটার প্যারেন্ট কে সেটা মনে রাখতে হবে কোনো একটা অ্যারেতে।

তুমি যদি বিএফএস ব্যবহার করে সমাধান করো তাহলে সেটাকে বলা হয় **এডমন্ড কার্প** অ্যালগোরিদম। ফোর্ড-ফুলকার্সন অ্যালগোরিদমে শুধু পাথ খুঁজে বের করার কথা বলে হয়েছে, সেটা যেকোনোভাবে বের করা যেতে পারে, আর এডমন্ড-কার্প বিএফএস ব্যবহার করে কাজটা করতে বলা হয়েছে অর্থাৎ ফোর্ড-ফুলকার্সন ইমপ্লিমেন্ট করার একটা উপায় হলো এডমন্ড কার্প অ্যালগোরিদম যার কমপ্লেক্সিটি  $O(VE^2)$ ।

এখন তোমাদের জন্য চিন্তা করার মত কয়েকটা প্রশ্ন:

**প্রশ্ন ১:** আমাদের প্রবলেমে সোর্স এবং সিংক ছিলো একটা। কিন্তু গ্রাফে একাধিক নোড দিয়ে পানি প্রবেশ করলে এবং একাধিক নোড দিয়ে পানি বের হয়ে গেলে কিভাবে অ্যালগোরিদমটা পরিবর্তন করবে?

**প্রশ্ন ২:** যদি প্রতিটা নোডের কিছু ক্যাপাসিটি থাকে, অর্থাৎ একটা নোড দিয়ে কত সর্বোচ্চ ফ্লো পাঠানো যাবে সেটা নির্দিষ্ট করা থাকে তাহলে কিভাবে সমাধান করবে?

**প্রশ্ন ৩:** দুটি নোডের মধ্যে একাধিক এজ থাকলে কি করবে?

**প্রশ্ন ৪:** দুই বন্ধু একই নোড থেকে যাত্রা শুরু করে একই গন্তব্যে পৌঁছাতে চায় কিন্তু দুইজনেই চায় ভিন্ন ভিন্ন রাস্তা ব্যবহার করে যেতে, তারমানে একই এজ কখনো ২জন ব্যবহার করতে পারবে না। গ্রাফটি দেয়া হলে তুমি কি বলতে পারবে এরকম ২টি পথ আছে নাকি? এ ধরণের পথকে **এজ ডিসজয়েন্ট পাথ** বলে।

এই প্রশ্নগুলো নিয়ে আলোচনা করা হয়েছে **পরের পর্বে**।

প্রোগ্রামিং কনটেস্টে ম্যাক্সিমাম ফ্লো প্রবলেম এ কোডিং এর থেকে অনেক কঠিন হলো গ্রাফটা কিভাবে বানাতে হবে সেটা বের করা, তাই অনেক সমস্যা সমাধান করে প্র্যাকটিস করতে হবে। শুরু করার জন্য কয়েকটা প্রবলেম দিয়ে দিলাম:

[http://lightoj.com/volume\\_showproblem.php?problem=1153](http://lightoj.com/volume_showproblem.php?problem=1153)

[http://lightoj.com/volume\\_showproblem.php?problem=1155](http://lightoj.com/volume_showproblem.php?problem=1155)<http://uva.onlinejudge.org/external/110/11045.html>

হ্যাপি কোডিং!

(এখন থেকে কোনো লেখায় সি++/জাভায় সোর্স-কোড দেয়া হবে না এবং আগের লেখাগুলোর সি++ সোর্স-কোডগুলোও ধীরে ধীরে সরিয়ে নেয়া হবে, তুমি যদি অ্যালগোরিদমটা বুঝে থাকো তাহলে নিজেই কোড লিখতে পারবে। আর বুঝতে সমস্যা হলে বা বিশেষ কোনো কারণে কোড চাইলে সরাসরি আমার সাথে [যোগাযোগ কর](#))

ফেসবুকে মন্তব্য

0 comments

0 comments

Sort by Oldest

Add a comment...

Facebook Comments plugin

Powered by Facebook Comments



Posted in অ্যালগরিদম/প্রবলেম সমাধান, প্রোগ্রামিং ? Tagged গ্রাফ, গ্রাফ থিওরি, ম্যাক্স ফ্লো

30,263 times read (excluding bots)

◀ কোয়ান্টাম কম্পিউটার - ২ (শক্তি এবং সীমাবদ্ধতা)

গ্রাফ থিওরিতে হাতেখড়ি-১২ – ম্যাক্সিমাম ফ্লো (২) ▶

4 thoughts on "গ্রাফ থিওরিতে হাতেখড়ি ১২ – ম্যাক্সিমাম ফ্লো (১)"



নিরব

জানুয়ারি ৩১, ২০১৫ at ২:৫৪ pm

ভাইয়া, এডমন্ড কার্পের কমপ্লেক্সিটি হবে  $O(V^2E)$ । Dinic's অ্যালগোরিদম আরও কিছু টেকনিক ব্যবহার করে টাইম কমপ্লেক্সিটি  $O(V^2E)$  তে নিয়ে আসতে পারে।