



## শাফায়েতের ব্লগ

প্রোগ্রামিং, অ্যালগরিদম, ব্যাকএন্ড ইঞ্জিনিয়ারিং

Home

অ্যালগরিদম নিয়ে যত লেখা!

আমার সম্পর্কে...

# গ্রাফ থিওরিতে হাতেখড়ি ৭:টপোলজিকাল স্ট

📅 অক্টোবর ৬, ২০১১ by Shafaet Ashraf



### (অন্যান্য পর্ব)

মনে কর তোমার হাতে কিছু কাজের একটা তালিকা আছে, কাজগুলো অবশ্যই শেষ করতে হবে। কাজগুলো হলো অফিসে যাওয়া, সকালে নাস্তা করা, টিভিতে খেলা দেখা, কিছু ই-মেইলের উত্তর দেয়া, বন্ধুদের সাথে ডিনার করা ইত্যাদি। কাজগুলো কিন্তু আপনি যেকোনো অর্ডারে করতে পারবেনা, কিছু শর্ত মানতে হবে। যেমন অফিসে যাবার আগে নাস্তা করতে হবে, খেলা দেখার আগে অফিসে যেতে হবে, ডিনারে বসার আগে ইমেইলের উত্তর দিতে হবে।

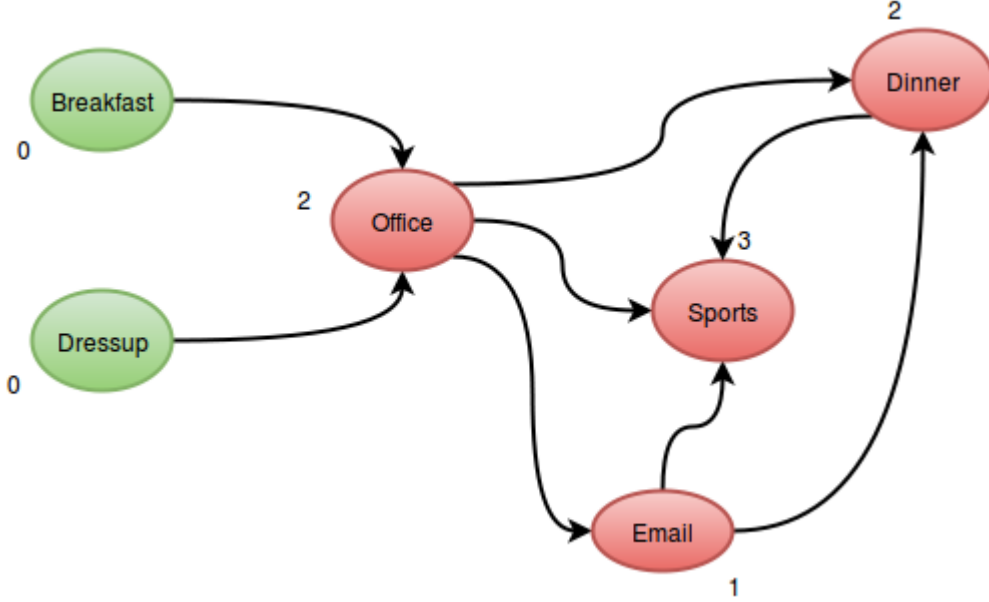
তুমি শর্তগুলোর তালিকা করে ফেললে:

১. সকালের নাস্তা —> অফিস (ক—>খ এর মানে হলো 'খ' কাজটি করার আগে 'ক' কাজটি করতে হবে)
২. সুট-টাই পড়া —> অফিস
৩. অফিস —> ইমেইল
৪. অফিস —> ডিনার
৫. অফিস —> খেলা
৬. ইমেইল —> ডিনার
৭. ইমেইল —> খেলা
৮. ডিনার —> খেলা

তুমি এখন কোন কাজ কখন করবে? উল্টাপাল্টা অর্ডারে করলে তোমার কাজ ভন্ডুল হয়ে যাবে, ইমেইল না করে খেলা দেখতে বসলে তুমি ক্লায়েন্ট হারাবে, তাই অর্ডারিং খুব জরুরি।

এটা একটি "টাস্ক শিডিউলিং" প্রবলেম। কোন কাজের পর কোন কাজ করতে হবে সেটা আমাদের বের করতে হবে। অর্থাৎ এটা এক ধরনের সর্টিং যাকে টপোলজিকাল সর্টিং বলে। আমরা এ টিউটোরিয়ালে এজ সরিয়ে বা ইনডিগ্রী কমিয়ে টপসর্ট বের করবো।

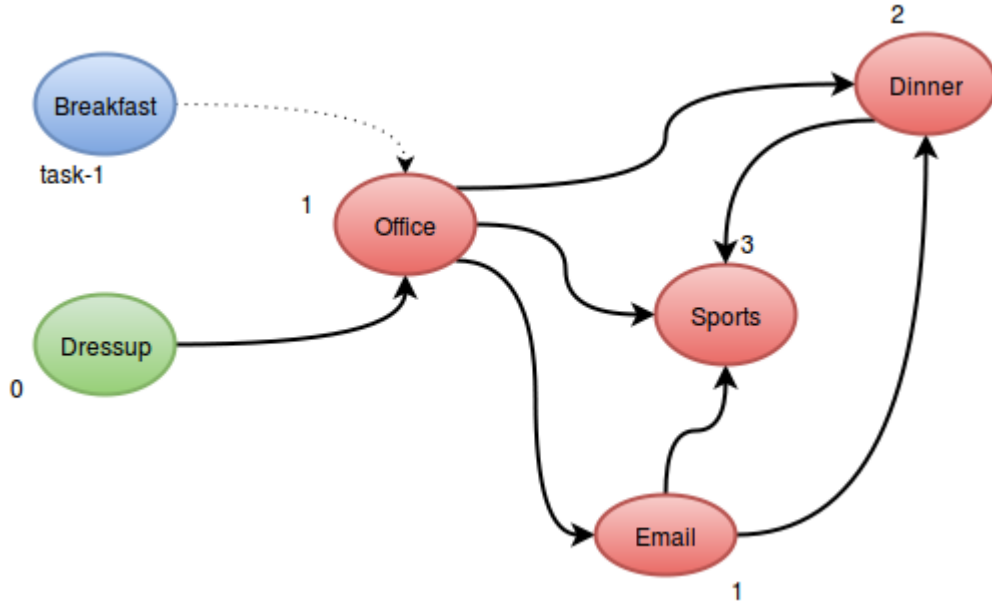
আমরা প্রথমেই সমস্যাটাকে নিচের গ্রাফ দিয়ে মডেলিং করবো:



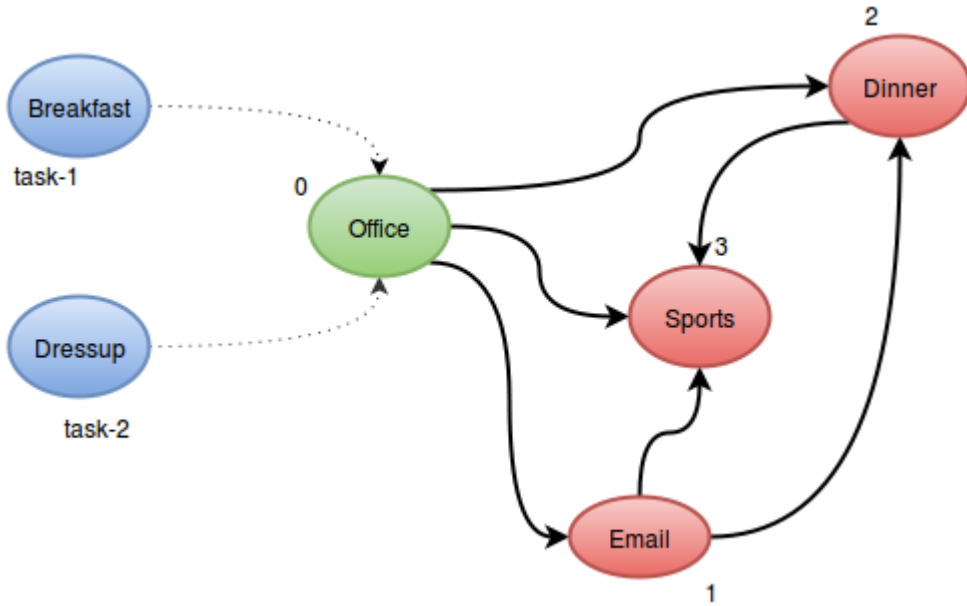
উপরের ছবিতে প্রতিটা কাজ একটি করে নোড দিয়ে দেখানো হয়েছে। ব্রেকফাস্ট থেকে অফিসের দিকে তীরচিহ্ন দিয়ে বুঝানো হচ্ছে যে অফিসে আসার আগে ব্রেকফাস্ট করতে হবে। উপরের ৮টি শর্ত ছবিতে ৮টি ডিরেক্টেড এজ দিয়ে দেখানো হয়েছে।

প্রতিটি নোডের পাশে ছোট করে কিছু সংখ্যা দেখতে পাচ্ছে। যেমন অফিসের সাথে ০, ডিনারের সাথে ২ ইত্যাদি। এগুলো দিয়ে বুঝাচ্ছে একটি কাজ অন্য কয়টি কাজের উপর নির্ভরশীল। যেমন ডিনারের আগে তোমাকে অফিস, ইমেইল এই ২টা কাজ করতে হবে, ডিনার নোডটিতে ২টি তীরচিহ্ন প্রবেশ করেছে আর আমরা পাশে লিখে দিয়েছি "২"। ঠিক এভাবে ইমেইলের পাশে লেখা হয়েছে ১। এ সংখ্যাগুলোকে indegree বলা হয়।

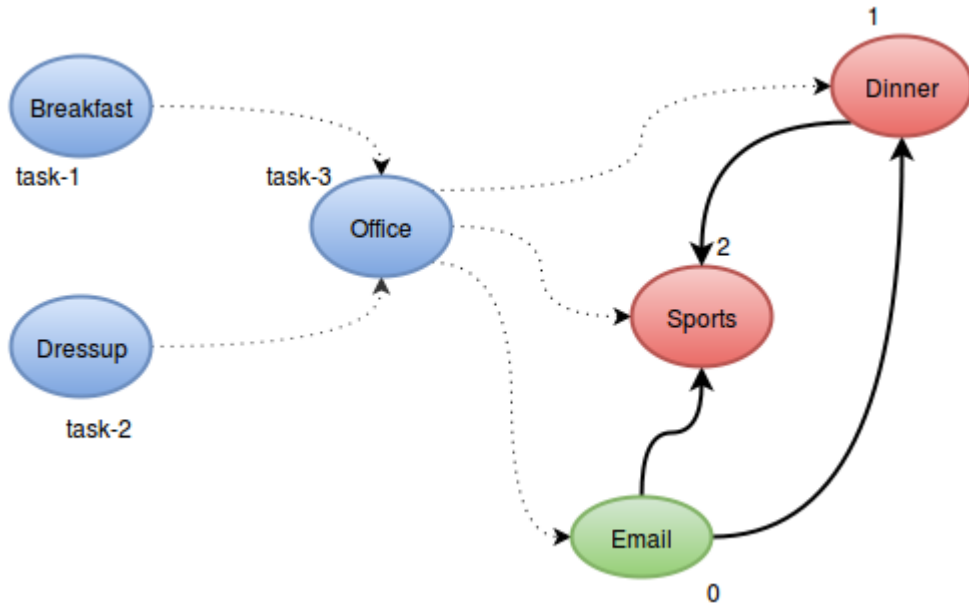
লক্ষ্য কর ব্রেকফাস্ট এবং ড্রেসআপ কোনো কাজের উপর নির্ভরশীল নয়, তাই তাদের পাশে ০ লেখা হয়েছে। তারমানে আমরা এ দুটি কাজের **যেকোনোটা** দিয়ে দিন শুরু করতে পারি। মনে করি তুমি নাস্তা আগে খেতে চাও। নাস্তা খেয়ে নেবার পর যেসব কাজ ব্রেকফাস্টের উপর নির্ভরশীল ছিল তারা আর সেটার উপর নির্ভরশীল থাকলোনা, গ্রাফটা হয়ে গেল এরকম:



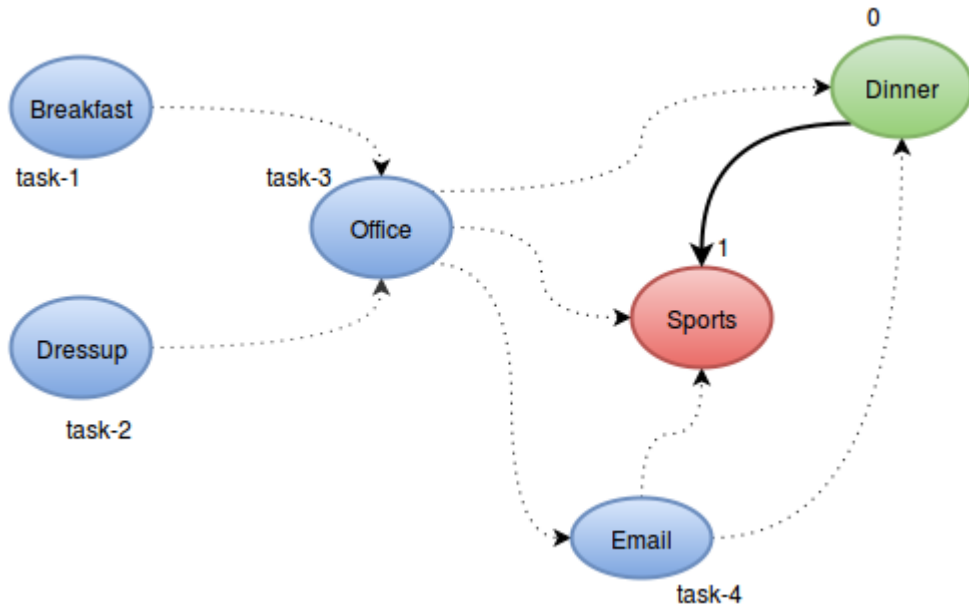
ব্রেকফাস্ট থেকে অফিসের তীরচিহ্ন সরিয়ে দিয়েছি। এখন অফিস আর মাত্র ১টি কাজের উপর নির্ভরশীল(আগে ছিল ২টির উপর)। এবার তোমাকে ড্রেসআপ করতে হবে, কারণ এখন একমাত্র এই কাজটিই কারো উপর নির্ভরশীল নয়:



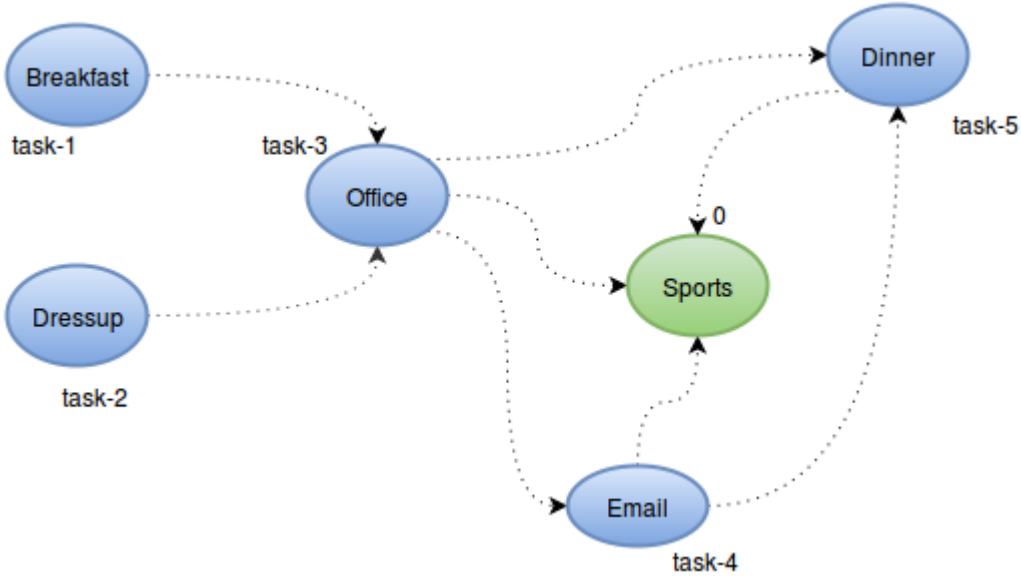
ড্রেসআপ থেকে অফিসের তীরচিহ্ন সরিয়ে দেয়া হয়েছে, আর কোনো কাজ নেই, এবার তুমি অফিসে যাবার জন্য প্রস্তুত। অফিসে যাবার উপর যারা নির্ভরশীল তাদের তীরচিহ্নগুলো এখন সরিয়ে দিতে পারি:



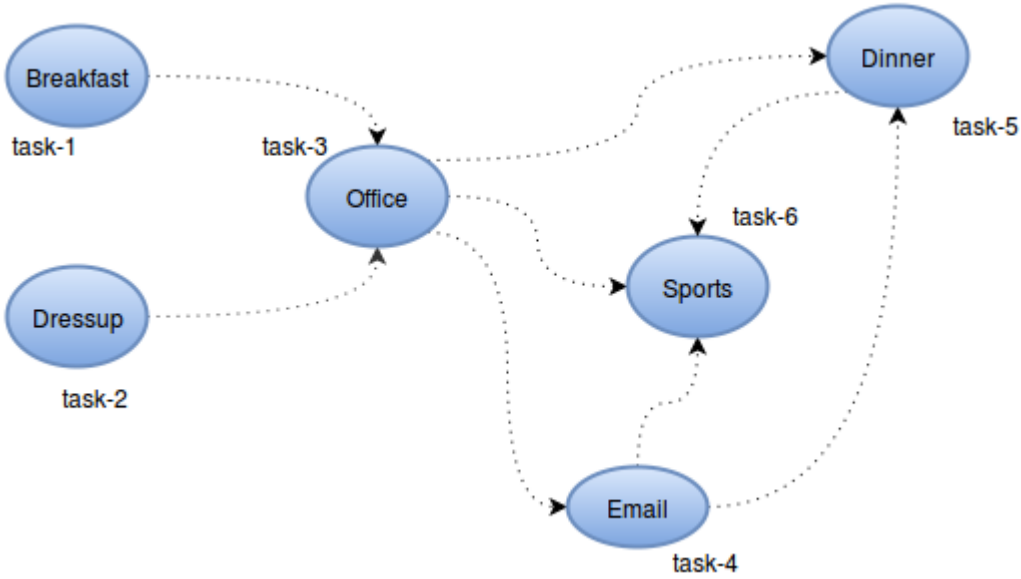
এখন ইমেইল “নোড” এর নির্ভরশীলতা ০ হয়ে গিয়েছে:



এরপর ডিনার:



সবশেষে খেলা দেখতে বসা:



এখন তুমি কাজের অর্ডারিং পেয়ে গিয়েছো, নান্সা করা, অফিসের পোষাক পড়া, অফিসে যাওয়া, ইমেইল করা, ডিনার করা, খেলা দেখা।

এটাকেই বলা হয় টপোলজিক্যাল সর্ট(topological sort) বা টপসর্ট। মূলত কাজের অর্ডারিং খুঁজে বের করতে এই অ্যালগোরিদমটি ব্যবহার করা হয়। কম্পিউটার তার অভ্যন্তরে বিভিন্ন কাজের অর্ডারিং ঠিক করতে টপসর্ট ব্যবহার করে। অনেক রিয়েল লাইফ প্রয়োগ থাকায় টপসর্ট কম্পিউটার সায়েন্সে খুবই গুরুত্বপূর্ণ একটি টপিক। টপসর্ট বের করার আরেকটি পদ্ধতি আছে যার নাম “ডেপথ ফার্স্ট সার্চ”, এটা নিয়ে আলোচনা করেছি এই লেখায়।

উপরের অ্যালগোরিদমটি  $O(n^2)$  এ কাজ করে। একটি গ্রাফের অনেকগুলো সর্টেড অর্ডার থাকতে পারে, যেমন উপরের সমস্যায় তুমি নান্সা খাবার আগে জামা-কাপড় পরতে পারতে। তোমাকে লেক্সিকোগ্রাফিক্যালি ছোটটা প্রিন্ট করতে বলতে পারে অথবা যেটা ইনপুটে আগে আছে সেটাকে আগে প্রিন্ট করতে বলতে পারে, আশা করি এসব কন্ডিশন সহজে হ্যান্ডল করতে পারবে।

ইমপ্লিমেন্টেশন নিয়ে তেমন কিছু বলার নেই। সবগুলো নোডের indegree বের করবে। তারপর যার indegree শূন্য তার সাথে যাদের এজ আছে তাদের indegree ১ কমিয়ে দিবে, তারপর আবার খুঁজবে কার indegree এখন শূন্য। এক নোডকে কখনো ২বার নিবেনা। ছবিতে এজ উঠিয়ে দেখিয়েছি বুঝানোর জন্য, তোমার ম্যাট্রিক্স থেকে এজ উঠানোর দরকার নেই, indegree কমালেই চলবে।

অনেক সময় বলতে পারে যতরকম ভাবে topsort করা যায় সবগুলো বের করতে। তখন তোমাকে **backtracking** এর সাহায্য নিতে হবে।

নিচের সমস্যাগুলো সমাধান করে ফেলো:

<http://uva.onlinejudge.org/external/103/10305.html>(Easy, straight-forward, special judge)

<http://uva.onlinejudge.org/external/110/11060.html>(Easy)

<http://uva.onlinejudge.org/external/1/124.html>(Medium, All possible topsort)

<http://uva.onlinejudge.org/external/4/452.html>(Medium)

(অন্যান্য পোস্ট)

## ফেসবুকে মন্তব্য

0 comments

0 comments

Sort by **Oldest**

Add a comment...

Facebook Comments plugin

Powered by **Facebook Comments**



Posted in অ্যালগরিদম/প্রবলেম সলভিং, প্রোগ্রামিং ? Tagged গ্রাফ থিওরি

55,285 times read (excluding bots)

↑ গ্রাফ থিওরিতে হাতেখড়ি ৬: মিনিমাম স্প্যানিং ট্রি(ক্রসকাল অ্যালগরিদম)

বিটওয়াইজ্ সিভ(Bitwise sieve) ▶