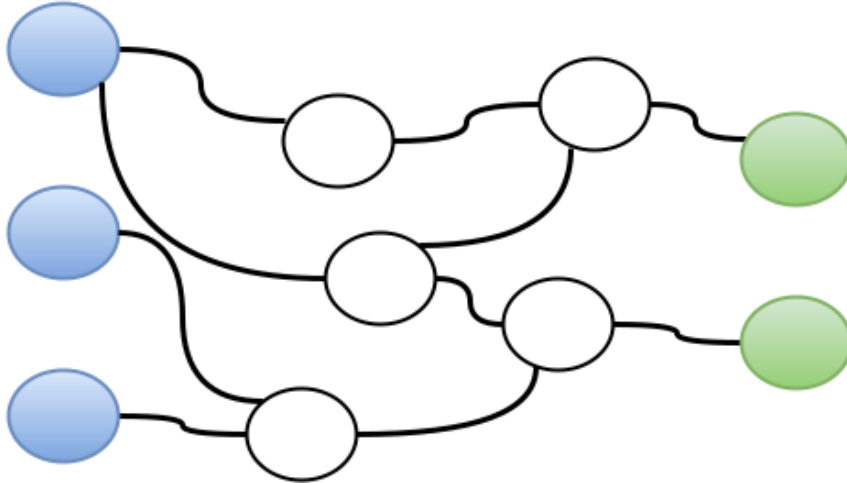


আগের পর্বে আমরা দেখেছি কিভাবে ফোর্ড-ফুলকারসন পদ্ধতি ব্যবহার করে ম্যাক্সিমাম ফ্লো বের করতে হয়। এই পর্বে ম্যাক্সিমাম ফ্লো সমস্যার সহজ কিছু ভ্যারিয়েশন দেখবো।

একাধিক সোর্স/সিংক:

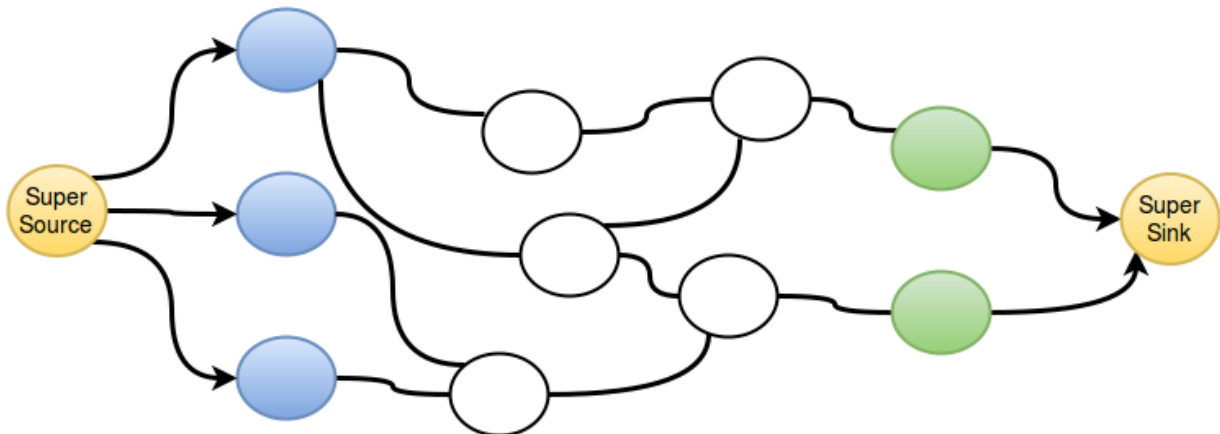
আগের পর্বে একটা প্রশ্ন করেছিলাম এরকম “আমাদের প্রবলেমে সোর্স এবং সিংক ছিলো একটা। কিন্তু গ্রাফে একাধিক নোড দিয়ে পানি প্রবেশ করলে এবং একাধিক নোড দিয়ে পানি বের হয়ে গেলে কিভাবে অ্যালগোরিদমটা পরিবর্তন করবে?” অর্থাৎ একাধিক সোর্স বা সিংক থাকলে কি করতে হবে সেটা জানতে চাওয়া হয়েছে।

চিত্র-১ এ বাম পাশের নীল নোডগুলো হলো সোর্স এবং ডানের সবুজ নোডগুলো হলো সিংক।



চিত্র -১: একাধিক সোর্স এবং সিংক সহ একটি গ্রাফ

এ ধরনের গ্রাফে এডমন্ড কার্প অ্যালগোরিদম প্রয়োগ করার সহজ উপায় হলো সুপার-সোর্স এবং সুপার সিংক বানিয়ে নেয়া। সুপার সোর্স হলো এমন একটা নোড যেটা সবগুলো সোর্সের সাথে ডিরেক্টেড এজ দিয়ে যুক্ত। ঠিক সেভাবে, সুপার সিংক প্রতিটি সিংকের সাথে ডিরেক্টেড এজ নিয়ে সংযুক্ত। এবং এই এজগুলোর ক্যাপাসিটি হবে অসীম বা ইনফিনিটি।

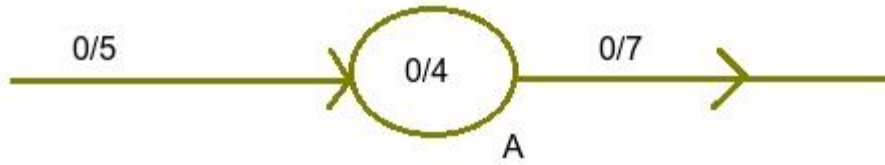


চিত্র-২: সুপার সোর্স এবং সুপার সিংক

চিত্র-২ তে সুপার সোর্স এবং সুপার সিংক দেখানো হয়েছে। ইনফিনিটি হিসাবে বেছে নিতে পারো সবগুলো এজের সম্মিলিত ক্যাপাসিটির থেকে বড় কোনো মানকে। এখন সাধারণ ফ্লো অ্যালগোরিদম ব্যবহার করেই এই গ্রাফে ম্যাক্সিমাম ফ্লো বের করতে পারবে।

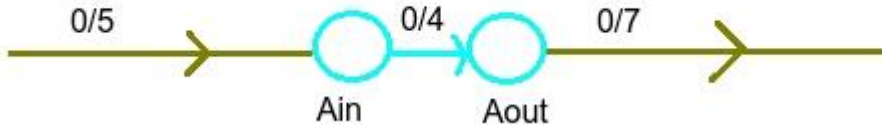
নোড ক্যাপাসিটি:

এতক্ষণ সবগুলো গ্রাফে এজের নির্দিষ্ট ক্যাপাসিটি ছিলো, নোডের ছিলো না। কিন্তু বাস্তবে অনেক সমস্যায় নোডের ক্যাপাসিটি থাকতে পারে। যেমন ধরো কোনো একটা দেশে প্রতিটা রাস্তার পাশাপাশি প্রতিটা শহরের নির্দিষ্ট গাড়ি ধারণ ক্ষমতা আছে, সেই দেশের গ্রাফ চিত্র-৩ এর মতো হতে পারে:



চিত্র-৩: নোড ক্যাপাসিটি

ছবিতে পুরো গ্রাফটা না একে শুধু একটা নোড আর ২টি এজ একেছি, নোডটাতে ঢোকার এজ এর ক্যাপাসিটি ৫, যে এজটি বাইরে চলে গেছে তার ক্যাপাসিটি ৭, এদিকে নোডের নিজের ক্যাপাসিটি ৪। আগে শেখা অ্যালগোরিদমে আমরা এজের ক্যাপাসিটির হিসাব রাখার জন্য একটা অ্যারে ব্যবহার করেছিলাম, এখনও আমরা সেই অ্যারেটা ব্যবহার করেই কাজ করতে পারবো, বুদ্ধিটা হলো নোডটাকে দুই ভাগে ভাগ করে ফেলা, এবং ভাগ দুটিকে নতুন এজ দিয়ে যোগ করে দেয়া। চিত্র-৪ দেখলেই পরিষ্কার হবে ব্যাপারটা:



চিত্র-৪

৪: A নোডটিকে দুইভাগ করা হয়েছে

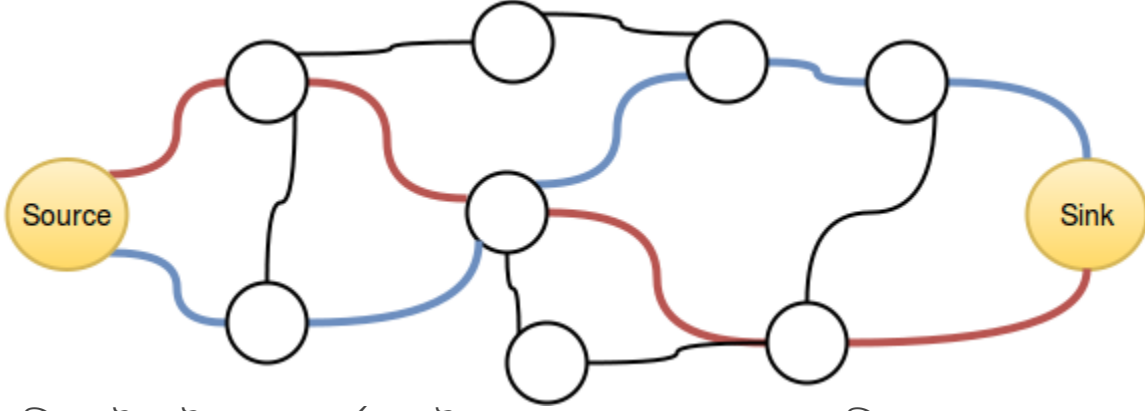
আমরা A নোডটা Ain এবং Aout এই দুটি নোডে ভাগ করেছি। এখন আসল গ্রাফ যতগুলো এজ A তে প্রবেশ করেছে সেগুলো প্রবেশ করবে Ain এ এবং আসল গ্রাফে যতগুলো এজ A থেকে বাইরে গিয়েছে সেগুলো এখন বাইরে যাবে Aout থেকে। Ain থেকে Aout এ একটা এজ প্রবেশ করবে যেটার ক্যাপাসিটি হবে এজ এর ক্যাপাসিটির সমান।

এখন নিশ্চিন্তে তুমি আগের অ্যালগোরিদম ব্যবহার করতে পারো। কোড লেখার সময় কিভাবে নোড দুইভাগ করবে, আনডিরেক্টেড গ্রাফের ক্ষেত্রে ব্যাপারটা কিরকম হবে সেগুলো চিন্তা করা তোমার কাজ!

এজ ডিসজয়েন্ট পাথ:

দুই বন্ধু একই নোড থেকে যাত্রা শুরু করে একই গন্তব্যে পৌছাতে চায় কিন্তু দুইজনেই চায় ভিন্ন ভিন্ন রাস্তা ব্যবহার করে যেতে, তারমানে একই এজ কখনো ২জন ব্যবহার করতে পারবে না। এধরণের

পথকে এজ ডিসজয়েন্ট পাথ বলে। তোমাকে বলতে হবে কোনো একটা গ্রাফে দুটি এজ ডিসজয়েন্ট পাথ আছে নাকি। চিত্র-৫ এ একটা উদাহরণ দেখানো হয়েছে:



ছবিতে দুইজনই বামের সোর্স নোডটা থেকে যাত্রা শুরু করে ডানের সিংক নোডে যেতে চায়। লাল এবং নীল রঙ ব্যবহার করে দুটি এজ-ডিসজয়েন্ট পাথ দেখানো হয়েছে।

সাধারণ ম্যাক্স-ফ্লো ব্যবহার করেই এজ ডিসজয়েন্ট পাথ বের করা যায়। শুরুর নোডকে সোর্স এবং গন্তব্য নোডকে সিংক ধরবে। এবার সবগুলো এজ এর ক্যাপাসিটি বানিয়ে দাও ১ এর সমান। এখন যদি তুমি সোর্স থেকে সিংকে দুই ফ্লো পাঠাতো পারো সেটার মানে হলো দুটি ডিসজয়েন্ট পাথ আছে। প্রতিটা এজের ক্যাপাসিটি ১ হওয়াতে ২ ফ্লো যে দুটি পথে গিয়েছে তাদের মধ্যে কমন এজ থাকা সম্ভব না।

ঠিক একই ভাবে তুমি একটা গ্রাফে সর্বোচ্চ কয়টা ডিসজয়েন্ট পাথ থাকা সম্ভব অথবা দুই বন্ধুর জায়গায় K টা বন্ধু থাকলে কি হতো বের করে ফেলতে পারবে।

এখন প্রশ্ন হলো তুমি যদি প্রতিটা রাস্তার নির্দিষ্ট দৈর্ঘ্য থাকে এবং ডিসজয়েন্ট পাথ দুটির মোট দৈর্ঘ্য মিনিমাইজ করতে চাও তাহলে ফ্লো এর অ্যালগোরিদমটা কিভাবে পরিবর্তন করবে? এটা বের করতে পারলে [uva 10806](#) সমস্যাটা সমাধান করে ফেলো, সমস্যাটার নামের ভিতরেই কিভাবে সমাধান করতে হবে বলা আছে!

আজকের পর্ব এখানেই শেষ। মিন-কাট এবং ম্যাচিং নিয়ে আলোচনার জন্য আরেকটা পর্ব অপেক্ষা করতে হবে। কনটেস্টে ম্যাক্স-ফ্লো প্রবলেমের কঠিন অংশ হলো গ্রাফটা কিভাবে তৈরি করবো, এজগুলো কিভাবে যোগ করবো, কোন এজের ক্যাপাসিটি কত এগুলো বের করা, এসব করার পর ফ্লো অ্যালগোরিদম চালিয়ে দেয়া সহজ কাজ। তাই তোমাকে প্রচুর প্র্যাকটিস করে এই জিনিসগুলো আয়ত্তে আনতে হবে।

কিছু প্রবলেম:

Down Went Titanic
Clever Naming Pattern
Diagonal Sum
হ্যাপি কোডিং!