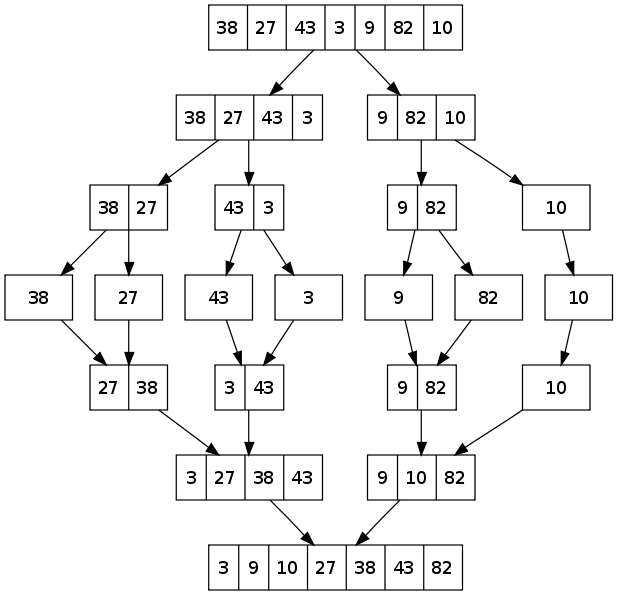
**মার্জ সর্ট বা জোড়া বাছাই**  
  
কুদ্দুস পড়ায় লবডঙ্কা হলেও কাজের বুদ্ধি প্রখর। বাবা দানু মিঞা সওদাগরের খাতুনগঞ্জের আড়তে বসতে হয়না এখনো, কিন্তু সেখানকার হালচাল ছোটবেলা থেকে দেখে আসাতে এই ধরণের কাজগুলো সহজে করে ফেলতে পারে। কেবল পরীক্ষার খাতাতেই কিছু লিখতে ইচ্ছে করে না। এই নিয়ে ২ বার ফেল করে ক্লাস এইটেই আটকে আছে।  
  
যাহোক, কুদ্দুসের মাথায় এলো, এতো ঝামেলা না করে কাজটাকে ছোট ছোট অংশে ভাগ করে ফেলা যাক। যেমন, আগের উদাহরণের তালিকাটা দেখা যাক। (৬৩, ৫৫, ৬৫, ৫২, ৭১, ৫৬) এই তালিকাটা এক বারে সাজানো কঠিন। তাই কুদ্দুস প্রথমেই এই তালিকাকে দুই ভাগে ভাগ করে ফেললো (৬৩, ৫৫, ৬৫) আর (৫২, ৭১, ৫৬)।  
  
বুদ্ধিটা হলো, এই দুইটা তালিকাকে প্রথমে সাজিয়ে ফেলবে, তার পর এদের দুই তালিকাকে একসাথে জোড়া লাগাবে।  
(৬৩, ৫৫, ৬৫) তালিকাটাকে কীভাবে সাজাবে? একই রকম, দুই ভাগে ভাগ করে ফেলা হলো। মাঝের জনকে তো আর কাটা যায় না, তাই তালিকাটা না হয়, (৬৩, ৫৫) আর (৬৫) এভাবে ভাগ হলো।  
  
এবার তো প্রথম তালিকাটা, মানে (৬৩, ৫৫) কে সাজানো সোজা, এদের জায়গা বদল করেই পাওয়া গেলো (৫৫, ৬৩)। তার সাথে (৬৫) এই তালিকাটাকে জোড়া লাগাবো কীভাবে? দুই পাশে দুই তালিকার ছাত্রদের দাঁড় করালো কুদ্দুস। তার পর দুই তালিকার সামনের মাথায় যারা আছে, তাদের তুলনা করলো, যে ছোট, তাকে নেয়া হলো। তাই প্রথমে ৫৫ আর ৬৫ এর তুলনা করে নেয়া হলো ৫৫কে। তার পরে ৬৩ আর ৬৫ এর তুলনা করে ৬৩কে, আর এর পর যেহেতু প্রথম তালিকা শেষ, তাই দ্বিতীয় তালিকার সবাইকে পরপর নিয়ে পাওয়া গেলো (৫৫, ৬৩, ৬৫)।  
  
ব্যাস, শুরুর তালিকাটা গুছানো হয়ে গেলো, পরের ৩ জনের তালিকাটাও একইভাবে গুছিয়ে পাওয়া গেলো (৫২, ৫৬, ৭১)।  
  
এবার কুদ্দুসের হাতে দুটো দল, (৫৫, ৬৩, ৬৫), আর (৫২, ৫৬, ৭১)। এদেরকে জোড়া লাগানোর কাজ শুরু করলো কুদ্দুস।  
  
(**৫৫**, ৬৩, ৬৫), (**৫২**, ৫৬, ৭১), [ ] -> (৫৫, ৬৩, ৬৫), (৫৬, ৭১), [**৫২**] (৫৫ আর ৫২ এর মধ্যে ৫২ ছোট)  
  
(**৫৫**, ৬৩, ৬৫), (**৫৬**, ৭১), [৫২] -> (৬৩, ৬৫), (৫৬, ৭১), [৫২, **৫৫**] (৫৫ আর ৫৬ এর মধ্যে ৫৫ ছোট)  
  
(**৬৩**, ৬৫), (**৫৬**, ৭১), [৫২, ৫৫]->(৬৩, ৬৫), (৭১), [৫২, ৫৫, **৫৬**]  
  
(**৬৩**, ৬৫), (**৭১**), [৫২, ৫৫, ৫৬] -> (৬৫), (৭১), [৫২, ৫৫, ৫৬,**৬৩**]  
  
(**৬৫**), (**৭১**), [৫২, ৫৫, ৫৬,৬৩]-> ( ), (৭১), [৫২, ৫৫, ৫৬, ৬৩, **৬৫**]  
  
( ), (**৭১**), [৫২, ৫৫, ৫৬,৬৩]-> ( ), ( ), [৫২, ৫৫, ৫৬, ৬৩, ৬৫, **৭১**]  
  
ব্যাস, গুছানো শেষ। আর এতে ঝামেলাও কম হলো। ঠিক কতোটা কম, কুদ্দুস নিজেও টের পায়নি, কিন্তু আমরা অংক করে দেখতে পারি, ক জন ছাত্র থাকলে তাদের সাজাতে সময় লাগবে ক x log(ক) সময়, যা আগের দুটো পদ্ধতির চাইতেই অনেক কম।  
মার্জ সর্টের বিভিন্ন ধাপের আরেকটি উদাহরণ দেখা যাক নিচের ছবিতে (সূত্রঃ উইকি),



9 July 2011 [0 Comments](http://www.acmsolver.org/bangla/?p=1861#respond)

[**প্রোগ্রামিং এর শৈল্পিক জগৎ-২**](http://www.acmsolver.org/bangla/?p=1861)

লেখক: শাফায়েত  
লেখকের ওয়েব: [http://www.shafaetsplanet.com/planetcoding/](http://www.shafaetsplanet.com/planetcoding/?p=211)

ধরা যাক বাংলাদেশের ১৫কোটি মানুষের নাম নিয়ে একটি তথ্যকেন্দ্র তৈরি করা হলো। নামগুলো বর্ণক্রমানুসারে সাজানো। এখন সেখান থেকে একটি নাম খুজে বের করতে হবে। সাধারণভাবে একটি নামের সাথে মিলিয়ে খুজে বের করতে সাধারণ কম্পিউটারের বেশ সময় লাগবে। কিন্তু আপনি সর্বোচ্চ ২৪বার মিলিয়ে নামটি খুজে বের করতে পারবেন। এখানেই প্রোগ্রামিং এর সৌন্দর্য। দৈনন্দিন জীবনের এমন বহু সমস্যা সমাধান করতে কম্পিউটার বিজ্ঞানী,প্রোগ্রামার,গণিতবিদরা বের করেছেন বুদ্ধিদীপ্ত সব অ্যালগোরিদম যা দিয়ে কম্পিউটারের কাজ সম্পাদনা করার সময়(execution time) বিষ্ময়কর ভাবে কমিয়ে আনা যায়।

বাইনারি সার্চ নামক একটি সহজ কিন্তু অসামান্য অ্যালগোরিদমের সাথে কম্পিউটার বিজ্ঞানের যেকোনো ছাত্র পরিচিত। তবে কিছুটা চিন্তা না করলে এটার ক্ষমতা অনুভব করা যায়না। মনে করি কিছু সংখ্যা ছোট থেকে বড় সাজানো আছে:

১ ১০ ২০ ৩০ ৪৫ ৫০ ৬০ ৯০ ১০০

এখন আমাকে খুজে বের করতে হবে ২০ সংখ্যাটি আছে নাকি। প্রথমেই ঠিক মাঝের সংখ্যাটি দেখি। মাঝে আছে ৪৫ যা ২০ থেকে বড়। তাহলে আমরা ৪৫ থেকে পরের সংখ্যা গুলো বাদ দিয়ে দিতে পারি কারণ তারা সবাই ২০ থেকে বড়[যেহেতু ৪৫ থেকে বড়]। তাহলে আমার কাছে থাকল:

১ ১০ ২০ ৩০

এখন মাঝের সংখ্যা ১০ যা ২০ থেকে ছোট।(সংখ্যা ৪টি হলে ২ বা ৩ তম সংখ্যার যেকোনোটাকে mid ধরা যাবে) তাই ১০ ও তার আগের সব সংখ্যা বাদ কারণ তারা সবাই ২০ থেকে ছোট। বাকি থাকল:

২০ ৩০

এবার আমরা মাঝের সংখ্যা হিসাবে ২০ কে পেয়ে যাব। তারমানে যতক্ষননা মাঝে কাঙ্খিত সংখ্যাটিকে মাঝখানে পাচ্ছি ততক্ষণ অর্ধেক করে বাদ দিতে থাকব।

ব্যপারটি অনেকটা এরকম,১০০ জন মানুষকে উচ্চতা অনুযায়ী সাজানো হলো,এখন নির্দিষ্ট উচ্চতার একজন মানুষ “ক” কে খুজতে আমরা প্রথমে মাঝের মানুষটির সাথে মিলিয়ে দেখব। “ক” এর উচ্চতা তার থেকে কম হলে ডানের সবাইকে বাদ দিয়ে দিব কারণ ডানের সবার উচ্চতা বেশি। যতজন মানুষ বাকি থাকল তাদের নিয়ে আবার একই ভাবে খুজব। প্রতিবার মানুষ সংখ্যা অর্ধেক হয়ে যাবে।  
উৎসাহীদের জন্য বাইনারি সার্চের একটি c++ কোড দিয়ে দিলাম:

//key is the value we are searching  
//low and high are array index  
//num is a global array  
void search(int low,int high,int key)  
{  
if(low>high) return;  
int mid=(low+high)/2;  
if(num[mid]<key)  
search(mid+1,high,key); //বামের সব মানুষ বাদ,মাঝের পরের জনই এখন সর্ববামের মানুষ  
else if(num[mid]>key)  
search(low,mid-1,key); //ডানের সব মানুষ বাদ,মাঝের আগের জনই এখন সর্বডানের মানুষ  
else  
{cout<<”Found at pos: “<<mid<<endl; return;} //পেয়েছি   
}

সংখ্যা ছাড়াও নামের ক্ষেত্রেও(string) এভাবে তুলনা করা যায়। ১০০টি নাম থাকলে প্রথমবার বাদ দেবার পর বাকি থাকবে ৫০টি,এরপর ২৫টি,এরপর ১২টি….। ২^৬৪ টি নাম থাকলে মাত্র ৬৪বার মিলিয়ে কাংখিত সংখ্যা খুজে বের করা যাবে!!!।

বাইনারি সার্চের প্রয়োগের ক্ষেত্র বিশাল। বিশেষ কিছু সমস্যা,এমনকি জ্যামিতির অনেক সমস্যার ফলাফল বাইনারি সার্চ দিয়ে বের করে ফেলা যায়,গাণিতিক ভাষায় একে bisection ও বলা হয়। binary search tree নামক একটি গুরুত্বপূর্ন ডাটা স্ট্রাকচার রয়েছে। কম্পিউটারে তথ্য বিভিন্নভাবে সাজিয়ে রাখা যায়,সহজ ভাষায় তথ্য সাজানো বিভিন্ন উপায়কেই ডাটা স্ট্রাকচার বলে। হাতে-কলমে যে ফলাফল বের করতে অনেক সময় লাগবে,একটি ছোট কোড লিখে সেটা সেকেন্ডের মাঝে করে ফেলা যায়। সামান্য গুতিয়ে প্রমান করা যায় nটি বস্তু থেকে কোনটাকে খুজে বের করতে বাইনারি সার্চে সর্বোচ্চ log2(n) বার মিলিয়ে দেখতে হবে,চাইলে বের করে দেখতে পারেন  ।

তবে অবশ্যই ডাটাগুলো সর্টেড হতে হবে,অর্থাত নির্দিষ্ট ক্রমানুসারে সাজানো থাকতে হবে। প্রথমবার সাজানোর সময় সামান্য কিছু সময় লাগবে। সর্টিং এর জন্য চমতকার কিছু অ্যালগোরিদম আছে যা খুব কম সময় ডাটা সাজিয়ে ফেলতে পারে।