جزوه جلسه اول داده ساختارها و الگوریتم

۲۸ شهریور ۱۴۰۰

فهرست مطالب

۱ آشنایی با الگوریتم ها

موضوع کلی درس درمورد روندهای بهینه برای حل مسائل با مقیاس نسبتا بزرگ در کنار تحلیل و طراحی این روندها است.
مسائل با مقیاس بزرگ با داده هایی نسبتا بزرگ نیز سروکار دارند. اما منظور از داده های بزرگ(Big Data) چیست؟ داده ها هنگامی Big Data تلقی میشوند که در یک کامپیوتر جا نشوند و برای ذخیره سازی و استفاده از آنها مجبور به استفاده از چند کامپیوتر یا سرور باشیم.
در مورد بهینگی یک الگوریتم میتوان به مواردی مانند پایین بودن زمان اجرا، کم بودن حافظه و منابع مصرفی، درستی الگوریتم برای داده های مختلف، کلی بودن الگوریتم، ساده بودن راه حل، خلاقانه بودن الگوریتم و مقیاس پذیر بودن آن اشاره کرد. منظور از مقیاس پذیری یک الگوریتم چیست؟ الگوریتمی را تصور کنید که برای ۵۰۰۰ ورودی به درستی کار میکند. حال اگر تعداد ورودی ها به ۱۰۰۰۰ تا افزایش یافت در اینصورت زمان اجرای الگوریتم و منابع مصرفی به چه مقدار تغییر میکنند؟ در این مثال اگر زمان اجرا و حافظه مورد استفاده دو برابر شوند، الگوریتم مقیاس پذیر محسوب میشود اما اگر الگوریتم برای ۱۰۰۰۰ داده اصلا کار نکند الگوریتم مقیاس پذیر محسوب نمیشود.
مفهوم بعدی Test of Time است. Test of Time به این معنیست که کاربردی بودن یک الگوریتم تنها توسط زمان ثابت میشود. یکی از دلایلی که امروزه از الگوریتم های چندین دهه قبل استفاده میکنیم همین مفهوم است.

۲ مثال های اولیه از الگوریتم های ساده

۱.۲ مسئله پیدا کردن قله یک آرایه یک بعدی

آرایه ای از اعداد به طول n را در نظر بگیرید. عضوی از آرایه را قله مینامیم اگر از دو عضو (برای عناصر ابتدا و انتها یک عضو) همسایه خود کوچکتر نباشد. آرایه مدنظر از ابتدا در دسترس نیست و برای فهمیدن هر عضو آن باید آنرا بپرسیم. هدف این است که با کمترین تعداد پرسش یک قله پیدا کنیم.
ساده ترین راه برای پیدا کردن قله پرسیدن تمامی اعضا و پیدا کردن قله است. برای اینکار نیاز به n پرسش داریم. راه حل دیگر این است که از یک سمت شروع به پرسش کنیم تا به یک قله برسیم. در این حالت نیز بد بدترین حالت نیاز به n پرسش داریم.
یکی از الگوریتم های بهینه برای پیدا کردن قله این است که سه عضو از میانه آرایه را بپرسیم. اگر عضو وسطی از بین این سه عضو قله بود که مسئله حل میشود. اگر این سه عضو به صورت صعودی (نزولی) مرتب شده بودند به سمت راست (چپ) حرکت کرده و مجددا سوال میپرسیم. اگر قله پیدا شد که الگوریتم به پایان میرسد و در غیر اینصورت به حرکت و پرسش روی اعضای آرایه ادامه میدهیم. این الگوریتم یا با پیدا یک قله قبل

از رسیدن به انتهای آرایه به پایان میرسد یا عضو انتهایی آرایه خود قله میشود.
حالت دیگری که بعد از پرسش سه عضو میانی با آن مواجه میشویم، حالتی است که عضو وسط از دو عضو دیگر کوچکتر باشد که در این حالت به یکی از دو سمت چپ یا راست حرکت میکنیم و مانند حالت صعودی/نزولی به پیدا کردن قله میپردازیم. در این الگوریتم حداکثر تعداد پرسش ها حدودا $n \div 2$ است.
اما بهینه ترین راه، نصف کردن آرایه است. در هر مرحله سه عضو میانی را میپرسیم که یا قله را پیدا کردیم یا بسته به حالت سه عضو که پرسیدیم یکی از دو نصفه چپ یا راست را انتخاب میکنیم و مجددا با پرسیدن سه عضو میانی به صورت بازگشتی مسئله را حل میکنیم. در اینصورت تعداد سوالهایی که میپرسیم در بدترین حالت به تعداد $3log_2n$ است. بدیهی است که برای nهای بزرگ داریم: $n \div 2 >> 3log_2n$