

۱ صرفاً مقایسه

فرض کنید یک آرایه از n عدد صحیح دارید و باید آرایه را به ترتیب صعودی مرتب کنید. بهترین و بدترین سناریوها را برای الگوریتم‌های مرتب‌سازی زیر مقایسه کنید: *Mergesort*, *Insertionsort*, *Bubblesort*. برای هر الگوریتم، بهترین و بدترین پیچیدگی‌های زمانی و همچنین توضیح مختصری در مورد چگونگی دستیابی به این موارد ارائه دهید. علاوه بر این، عملکرد این الگوریتم‌ها را از نظر پیچیدگی زمانی مقایسه کنید و مشخص کنید کدام الگوریتم‌ها را برای مرتب‌سازی آرایه‌های بزرگ توصیه می‌کنید.

۲ *Mergesort* به بد

به شما یک پیاده‌سازی از *MergeSort* داده می‌شود که از پیچیدگی فضای $O(n)$ به دلیل استفاده از یک آرایه کمکی در عملیات ادغام استفاده می‌کند. وظیفه شما بهینه‌سازی الگوریتم برای استفاده از پیچیدگی فضای $O(1)$ به جای آن است، بدون اینکه پیچیدگی زمانی $O(n \log n)$ را قربانی کنید.

- الگوریتم *MergeSort* اصلی را شرح دهید و توضیح دهید که چرا از پیچیدگی فضای $O(n)$ استفاده می‌کند.
- اصلاحی را برای الگوریتم *MergeSort* پیشنهاد کنید که پیچیدگی فضا را به $O(1)$ کاهش دهد و در عین حال پیچیدگی زمانی $O(n \log n)$ را حفظ کند. توضیح دهید که چگونه اصلاح شما به این امر می‌رسد.
- پیچیدگی زمانی و مکانی الگوریتم مرتب‌سازی ادغام شده خود را تجزیه و تحلیل کنید و آن را با الگوریتم اصلی مقایسه کنید

۳ سریع تر همیشه

مسئله مرتب سازی آرایه ای از n عنصر متمایز را به ترتیب غیر کاهشی در نظر بگیرید. با استفاده از مدل درخت تصمیم ثابت کنید که کران پایین برای این مسئله $O(n \log n)$ است.