

سوال ۲) الگوریتم Insertion Sort مناسب تر است زیرا در این الگوریتم، زمانی که عناصر تقریباً مرتب باشند یا سرعت در فاصله کمی از مقادیر انتهای خود باشند، زمان اجرای الگوریتم به  $O(n)$  نزدیک می شود که بسیار است.

سوال ۱) فرجات ثابت در تحلیل Big O ثابتی ندارد زیرا در تحلیل پیچیدگی و بیشتر به رفتار تابع وقتی ورودی بسیار بزرگ است بستگی دارد و توجه می کنیم، فرجات ثابت یعنی تعداد عملیات هابی که مستقل از اندازه ورودی انجام می شوند. در تعریف بارگذاری تابع افزایشی داریم که ثابت است. برای مثال اگر دو الگوریتم داشته باشیم یکی که عملیات ثابت انجام دهد و دیگری که عملیات ثابت با افزایش ورودی این تعداد عملیات تأثیر زیادی در سرعت نهایی اجرا ندارد. الگوریتم نخواهد داشت، به عبارت دیگر در نهایت اجرای الگوریتم نخواهد داشت. سرعت رشد تابع پیچیدگی ده آن مهم تر از فرجات ثابت است.

سوال ۴)  $O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2)$

$O(n!) < O(2^n) < O(n^2)$

سوال ۵) الگوریتم مرتب سازی Merge Sort از تقسیم کردن آرایه به بخش و سپس مرتب سازی مجدد بخش با استفاده از الگوریتم Merge Sort استفاده می کنند. بهترین و بدترین حالت این الگوریتم به دلیل تقسیم دائمی آرایه به دو قسمت و سپس ادغام برای  $O(n \log n)$  است.



سوال سوم)

الگوریتم اول: حلقه بیرونی با تقسیم ۱۰ به ۳ در هر تکرار، آنگاه با پیچیدگی

$O(\log n)$  می‌ماند. در کدام از حلقه‌های داخلی به اندازه  $n$  تکرار می‌شوند، بنابراین

پیچیدگی آن‌ها  $O(n)$  است. در مجموع پیچیدگی زمانی این الگوریتم برابر است با  $O(n \log n)$ .

الگوریتم دوم: حلقه اول به اندازه  $n \times 1000$  تکرار می‌شود که می‌توان آن را

$O(n)$  در نظر گرفت. حلقه‌ی دوم، هر تکرار  $n$  بار اجرا می‌شود که می‌توان

پیچیدگی زمانی برابر است با  $O(n)$ .

الگوریتم سوم: این الگوریتم شامل ۲ حلقه است که هر کدام به صورت مستقل

اجرا می‌شوند. پیچیدگی حلقه اول  $O(N)$  و حلقه دوم  $O(M)$  است. در نتیجه مجموع آن‌ها  $O(N+M)$  است.

الگوریتم چهارم: حلقه‌ی بیرونی به اندازه  $O(n)$  اجرا می‌شود. حلقه‌ی داخلی به

دو اندازگی پذیرای کند انجام می‌دهد.  $O(\log n)$  پیچیدگی زمانی  $O(n \log n)$

الگوریتم ۵: ۳ حلقه دارد که است. پیچیدگی زمانی برای هر حلقه به ترتیب

$O(n)$  و  $O(n)$  و  $O(n)$  است.

بنابراین پیچیدگی زمانی الگوریتم برابر  $O(n^3)$  است.