بسم تعالی مهربان

Subject ۹۸۴۱۱۴۳۲
Date

محمد عرفان زارع زردینی

کوئیز ۱۷

س۱

در insertion sort، طبق اینکه همه عناصر یکسان اند، حالت best case رخ می دهد و هیچگاه داخل while نمی شود (already sorted)، است. پس O(n) می باشد. حال اما Heap sort، در بهترین حالت o(nlogn) است که همیشه ثابت. merge sort نیز در بهترین حالت ممکن آن nLog(n) است که بازهم همیشه ثابت و البته تمام مراحل (و همه مراحل انجام می شود) تقسیم‌ها و combine ها و divide ها هم با وجود یکسان بودن عناصر انجام می شود. selection sort هم که در بین گزینه‌ها بدترین حالت را دارد و حتی در best case هم O(n²) است و یکسان بودن عناصر روی آن تاثیری ندارد و در مراحلش دائما می گردد و هر خانه را با خانه‌های جلوترش مقایسه می کند. پس A گزینه درست (insertion sort)

س۲

worst case
- A. merge sort: O(nlogn) → logn تقسیم کردن × n ادغام کردن و sort آن
- B. bubble sort: O(n²) → همه عناصر با هم مقایسه → در تکرار آخر در هر sort ← n×(n-1) مقایسه
- C. Quick sort: O(n²) → مجبور به طی کردن همه مراحل شود n(n-1)
- D. selection sort: O(n²)

پس گزینه A، بهترین حالت (merge sort)

س۳ در روش merge sort در هر حال همه مراحل Divide، conquer و combine انجام می شود پس در نتیجه همیشه اردر آن یکسان است و چون logn تقسیم و با n صورت می شود O(nlogn) است.

پس
best case = O(nLogn)
average case = O(nLogn)
worst case = O(nLogn)

PAPCO

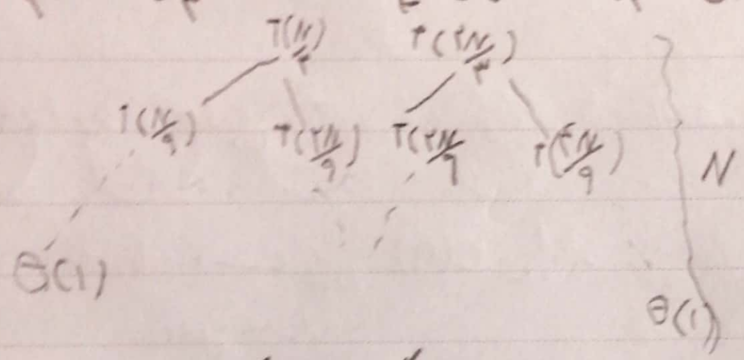
Subject

Date

س۴

با توجه به اینکه به یک بخش $N/3$ تقسیم شده و بخش دیگر $2N/3$ است پس داریم:

$$T(N)=T(N/3)+T(2N/3)+N$$



پس جواب ما می‌شود $T(N)=\theta(N\times(\log_{3/2} N))$ که گزینه D است

س۵ ما فشار: Heap Sort بدین گونه است که ابتدا یک Heap را ساخته و عملیات های ترتیبی زیر را انجام می‌دهیم تا مرتب شود

الف) Root را با عنصر جدید الورود جابجا کرده سپس heapify را روی Root و زیر درخت از سایر heap یکی می‌کنیم

الان پس در اینجا دیده می‌شود که دو عنصر آخر از همه بزرگتری با عنصر دو عناصر دیگر آنها به طور نا ترتیبی چیده شده‌اند

پس گویا ۲ بار بر روی این Heap صدا maxheapify زده شده و عمل روی آن رخ داده.