

۱ سوال اول ۰.۳

در این سوال، شما یک *DoublyLinkedList* را پیاده سازی خواهید کرد که از برخی عملیات زمان ثابت اضافی پشتیبانی می کند. هر گره x از یک *DoublyLinkedList*، یک اشاره گر $x.prev$ به گره قبل از آن در دنباله، علاوه بر یک اشاره گر $x.next$ به گره بعد از آن در دنباله نگه می دارد. یک *DoublyLinkedList* L یک اشاره گر به $L.tail$ ، آخرین گره در دنباله، علاوه بر $L.head$ ، اولین گره در دنباله را نگه می دارد.

برای این سوال، لیست های دارای پیوند دوگانه نباید طول خود را پیگیری کند (*keeping track*)
(آ) یک *DoublyLinkedList* دارید، الگوریتم هایی را برای اجرای عملیات های زیر، هر کدام در زمان $O(1)$ توصیف کنید.

- $insert\ first(x)$
- $insert\ last(x)$
- $delete\ first()$
- $delete\ last()$

(ب) دو گره x_1 و x_2 در یک *DoublyLinkedList* با نام L فرض کنید، که در آن x_1 قبل از x_2 قرار گرفته، یک الگوریتم زمان ثابت را برای حذف تمام گره ها از x_1 تا x_2 از L توصیف کنید و آنها را به عنوان یک *DoublyLinkedList* جدید برگردانید.

۲ سوال دوم ۰.۴

برای هر یک از سناریوهای زیر، یک الگوریتم مرتب سازی (از هر کدام از *insertionsort*، *selectionsort* یا *mergesort*) که بهترین کاربرد را دارد، انتخاب کنید و انتخاب خود را توجیه کنید. این را فراموش نکنید!

ارزش دلیل شما بیشتر از انتخاب شما خواهد بود.

هر الگوریتم *sort* ممکن است بیش از یک بار استفاده شود. اگر متوجه شدید که چندین نوع می‌توانند برای یک سناریو مناسب باشند، مزایا و معایب آن‌ها را شناسایی کنید و یکی را انتخاب کنید که برای برنامه کاربردی مناسب‌تر است. هر فرضی که می‌کنید را بیان و توجیه کنید. معایب "بهترین" باید با زمان *asymptotic* اجرا ارزیابی شود.

آ) فرض کنید به شما یک ساختار داده D داده شده است با n آیتم و از دو عملیات استاندارد پشتیبانی می‌کند: $D.getat(i)$ که در بدترین حالت $\theta(1)$ اجرا می‌شود و $D.setat(i, x)$ در بدترین حالت $\Theta(n \log n)$ اجرا می‌شود. الگوریتمی را انتخاب کنید تا موارد را در D به بهترین نحو در محل مرتب کند.

به در محل بودن این مرتب سازی توجه کنید.

ب) فرض کنید یک *static array* با نام A دارید حاوی نشانگرهایی به n شی قابل مقایسه دارید، جفت‌هایی که مقایسه شان زمان $\theta(\log n)$ برای مقایسه نیاز دارند. الگوریتمی را انتخاب کنید تا نشانگرها را در A به بهترین نحو مرتب کند تا اشیاء اشاره شده به ترتیب غیر کاهشی ظاهر شوند.

ج) فرض کنید یک آرایه مرتب شده A دارید که حاوی n عدد صحیح است که هر کدام در یک *word* *Machine* قرار می‌گیرند. حال فرض کنید شخصی $\log \log n$ عملیات *Swap* بین جفت آیتم‌های مجاور در A انجام می‌دهد به طوری که A دیگر مرتب نمی‌شود. الگوریتمی را انتخاب کنید تا اعداد صحیح در A را به بهترین نحو مرتب کند.

۳ سوال سوم ۰.۴

در یک آرایه، یک نابه جایی برابر یک جفت جایگاه است که عدد داخل جایگاه اولی بزرگتر از عدد داخل جایگاه دومی باشد. برای مثال در یک آرایه مرتب شده تعداد نابه جایی‌ها برابر صفر است و در یک آرایه مرتب شده برعکس این تعداد برابر $\binom{n}{2}$ است.

آ) ثابت کنید اگر تعداد نابه جایی‌های یک آرایه d باشد و الگوریتم *insertion sort* را روی آن اجرا کنیم با مرتبه زمانی $O(n + d)$ مرتب سازی انجام می‌شود.

ب) فرض کنید یک آرایه نامرتب به ما داده اند به طوری که هر کس با جایگاه اصلی آن حداکثر ۱۰ جایگاه فاصله دارد. برای مثال عدد کمینه حداکثر در اندیس ۱۰ است. (آرایه از اندیس صفر شروع

می شود.) با ارائه روش مناسب برای مرتب سازی ثابت کنید این کار در مرتبه زمانی $O(n)$ ممکن است.

۴ سوال چهارم ۰.۲

قطعه کدهای *Enqueue* و *Dequeue* را جهت *overflow* و *underflow* پیاده سازی کنید.

۵ سوال پنجم ۰.۲

آیا میتوان عملگر داینامیک *Insert* را روی یک لیست پیوندی یک طرفه در زمان $O(1)$ پیاده سازی کرد؟ در مورد عملگر *Delete* چطور؟