

ساختمان‌های داده و الگوریتم

تمرین چهارم - مرتب‌سازی و درهم‌سازی

محمد امانلو، فاطمه کرمی

تاریخ تحویل: ۱۴۰۲/۰۹/۲۶

نمره

.۱

الگوریتمی طراحی کنید که با گرفتن یک آرایه که در آن هر عنصر حداکثر ۲۰ عنصر سمت چپ خود دارد که از او بزرگ‌تر باشند، این آرایه را در زمان $O(n)$ مرتب کند.

نمره

.۲

یک الگوریتم با مرتبه زمانی $O(n \lg k)$ برای ادغام k لیست مرتب شده در یک لیست مرتب شده، که در آن n تعداد کل عناصر در همه لیست‌های ورودی است، ارائه دهید.

نمره

.۳

محمد فکر می‌کند که اگر روش زنجیره‌سازی مجزا (*separate chaining*) برای ساخت یک *hash table* را اینگونه تغییر دهد که در روش جدید هر لیست پیوندی در یک خانه آرایه به صورت مرتب شده باشد، نتیجتاً به کارایی (*performance*) بهتری خواهد رسید. در روش جدید محمد پیچیدگی زمانی برای جستجوی موفق، جستجوی ناموفق، درج و حذف را محاسبه کنید.

نمره

.۴

الگوریتم *Quick Sort* شامل دو فراخوان بازگشتی به خود است. پس از آنکه *Quick Sort*، *Partition* را فراخوانی می‌کند، زیرآرایه سمت چپ را به صورت بازگشتی مرتب می‌کند و سپس زیرآرایه سمت راست را به صورت بازگشتی مرتب می‌کند. فراخوانی بازگشتی دوم در *Quick Sort* واقعاً ضروری نیست و می‌توان با استفاده از یک ساختار کنترل تکرار شونده از آن اجتناب کرد. این تکنیک *tail recursion* نامیده می‌شود. نسخه زیر از *Quick Sort* را در نظر بگیرید که *tail recursion* را شبیه‌سازی می‌کند:

```
TAIL-RECURSIVE-QUICKSORT( $A, p, r$ )
1  while  $p < r$ 
2      // Partition and sort left subarray.
3       $q = \text{PARTITION}(A, p, r)$ 
4      TAIL-RECURSIVE-QUICKSORT( $A, p, q - 1$ )
5       $p = q + 1$ 
```

ثابت کنید که $\text{TAIL-RECURSIVE-QUICKSORT}(A, 1, A.\text{length})$ به درستی آرایه A را مرتب می‌کند.

نمره

.۵

الگوریتمی طراحی کنید که با گرفتن n عدد صحیح بین ۰ تا k پیش‌پردازشی روی ورودی انجام داده و سپس در زمان $O(1)$ با گرفتن دو عدد a و b مشخص کند که چه تعداد از اعداد ورودی در بازه $[a, \dots, b]$ هستند. پیش‌پردازشی که الگوریتم روی ورودی انجام می‌دهد باید با پیچیدگی زمانی $\Theta(n + k)$ باشد.

الگوریتمی با زمان اجرای $O(n)$ طراحی کنید که با گرفتن مجموعه S شامل n عدد یکتا و عدد مثبت k ، $k \leq n$ تا عددی را در S مشخص کند که نزدیک‌ترین به میانه S هستند.