**به نام یکتای هستی بخش**

*دانشگاه تهران، دانشکده‌ی مهندسی برق و کامپیوتر*

**ساختمان داده‌ها و الگوریتم‌ها**، نیمسال اول، سال تحصیلی ۹۴-۹۳

**تمرین شماره ۴** مهلت تحویل: یکشنبه ۹/۱۶ ، ساعت ۲۳:۵۵

الگوریتم‌هایی که برای سوالات زیر ارائه می‌دهید باید از بهترین مرتبه‌ی زمانی باشد.

۱.داده‌های زیر را به روش‌های merge sort، heap sort، quick sort و insertion sort مرتب کنید و مراحل آن‌ها را به صورت گام به گام و مختصر توضیح دهید. در مورد stable بودن الگوریتم‌ها نیز بحث کنید.

۴ ۱۲ ۴ ۱ ۲ ۳ ۱۲ ۹ ۲ ۱

۲.توضیح دهید در چه حالتی insertion sort بهتر از merge sort عمل خواهد کرد.

۳.تعداد n عدد در آرایه ای داریم. می‌خواهیم با کمترین هزینه تمامی اعدادی که قبل از عددی کوچکتر از خودشان قرار دارند را محاسبه کنیم. الگوریتمی مبتنی بر merge sort برای این کار ارائه دهید. سعی کنید الگوریتم جواب را به صورت inplace[۱] ارائه دهید.

۴.در یک آرایه به طول n عددی وجود دارد که بیشتر از n/2 ام بار در این آرایه تکرار شده است. الگوریتمی ارائه دهید که این عدد را پیدا کند. الگوریتم شما باید از مرتبه‌ی زمانی O(n) و مرتبه‌ی حافظه O(1) باشد.

۵.تعداد n عنصر در k لیست پیوندی داریم. که هر لیست به صورت صعودی مرتب شده است و اولین عنصر آن کوچک‌ترین عنصر است. الگوریتمی ارائه دهید که با بهترین مرتبه بتواند یک لیست پیوندی صعودی از عناصر این k دسته بسازد.

۶.اثبات کنید مرتبه ی تمامی الگوریتم‌های مرتب سازی مبتنی بر مقایسه در حوزه‌ی اعداد حقیقی حداقل O(nlogn) است.

۷.دنباله ای به طول n که شامل تمامی اعداد ۱ تا n است را با قطعه کد زیر مرتب می‌کنیم. (کد مربوط به merge sort است.)

Floorبه معنای کف و علامت .. به معنای بازه است.

```
function merge_sort(arr):
    n = arr.length()
    if n <= 1:
        return arr

    // arr is indexed 0 through n-1, inclusive
    mid = floor(n/2)
```

[۱]- الگوریتم inplace الگوریتمی است که در آن به حافظه‌ی اضافی زیادی نیاز نیست. به عبارتی الگوریتمی که مرتبه‌ی حافظه‌ی آن O(1) باشد، inplace است.

```
    first_half = merge_sort(arr[0..mid-1])
    second_half = merge_sort(arr[mid..n-1])
    return merge(first_half, second_half)

function merge(arr1, arr2):
    result = []
    while arr1.length() > 0 and arr2.length() > 0:
        if arr1[0] < arr2[0]:
            print '1' // for debugging
            result.append(arr1[0])
            arr1.remove_first()
        else:
            print '2' // for debugging
            result.append(arr2[0])
            arr2.remove_first()

    result.append(arr1)
    result.append(arr2)
    return result
```

الگوریتمی ارائه دهید که با گرفتن n و دنباله‌ی اعداد چاپ شده در الگوریتم (در قسمت مقایسه دو دستور print وجود دارد.) دنباله‌ی اصلی را تولید نماید.

به عنوان مثال برای دنباله‌ای به طول ۲ و دنباله‌ی چاپی ۱ باید لیست زیر تولید شود

[1, 2]

یا برای دنباله‌ای به طول ۴ و دنباله‌ی چاپی ۱۲۲۱۲ باید لیست زیر تولید شود.

[2, 4, 3, 1]

مرتبه‌ی الگوریتم شما باید از O(nlogn) باشد.

## نحوه‌ی تحویل:

لطفاً تمرین را به صورت اسکن شده در یک فایل فشرده با نام HW4[SID].zip در سایت درس آپلود کنید. SID پنج رقم آخر شماره‌ی دانشجویی شما است. یعنی اگر شماره دانشجوییتان ۸۱۰۱۹۲۰۰۰ است، نام فایلتان باید HW492000.zip باشد.

## نکات پایانی:

- ✓ به ازای هر روز تاخیر ۱۰ درصد از نمره‌ی تمرین را از دست خواهید داد. همچنین بیشترین میزان تأخیر مجاز ۵ روز است.
- ✓ در صورت مشاهده‌ی هرگونه تشابه نمره‌ی هر دو طرف ۱۰۰- منظور می‌گردد و در بار دوم نمره‌ی صفر برای درس منظور می‌گردد.
- ✓ در صورت وجود هر گونه سوال می‌توانید به فروم درس مراجعه کنید.