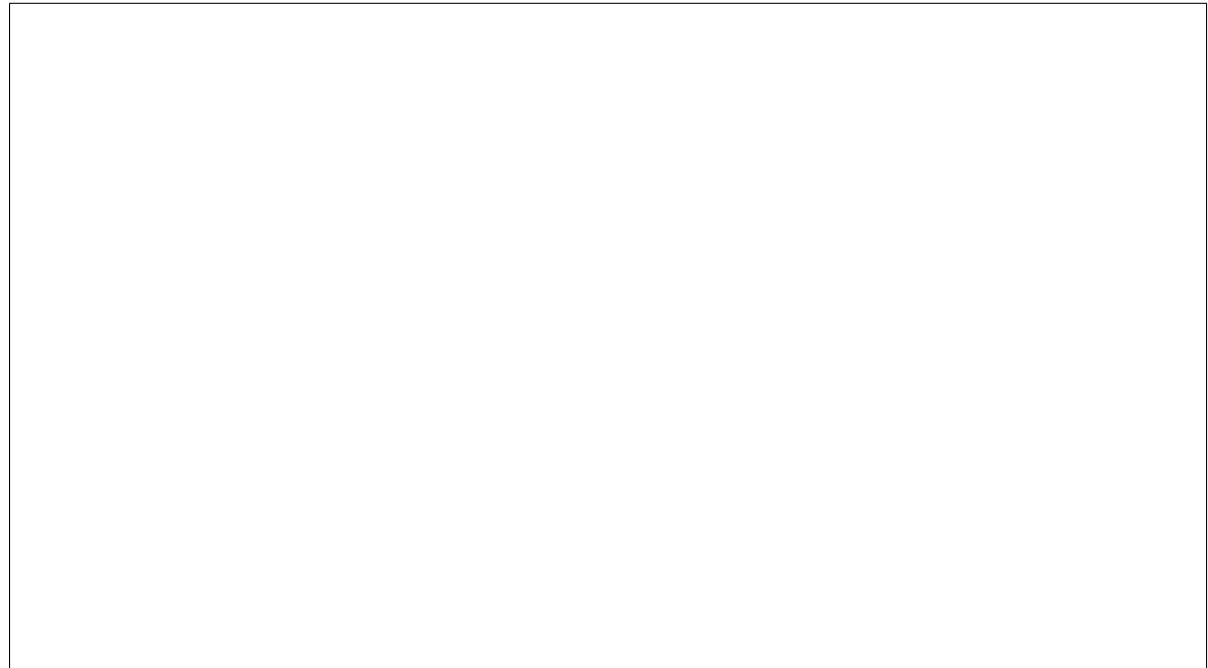


هر سوال را در محل در نظر گرفته شده پاسخ دهید. پاسخ های خارج از محل تصحیح نمیشوند. نام و شماره دانشجویی را روی تمام برگه ها بنویسید. شماره دانشجویی باید با اعداد لاتین نوشته شود. مهلت این تمرین شنبه ۱ آذر ماه است.

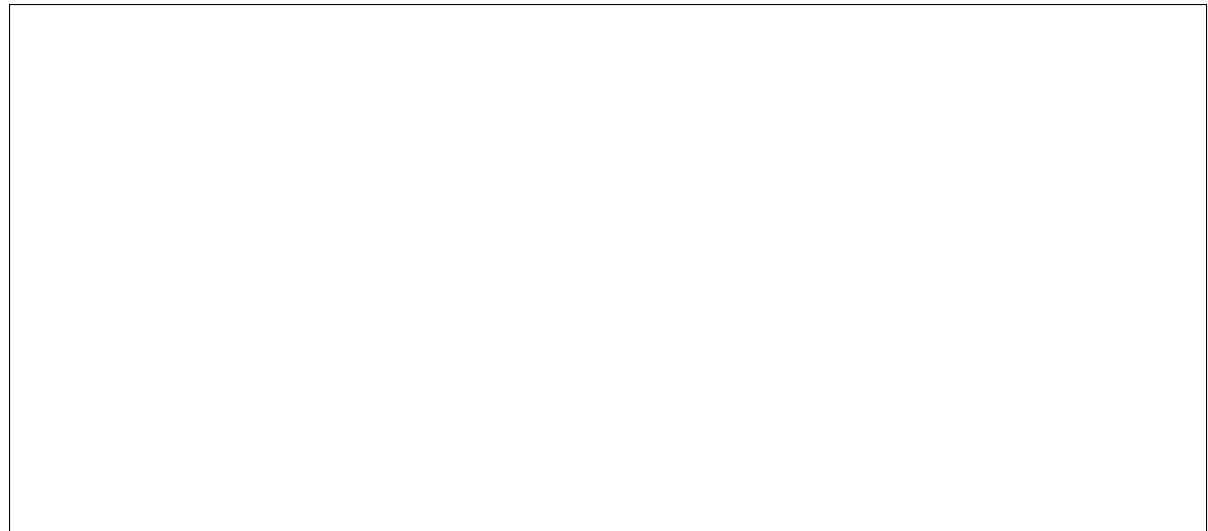
ملیکا نوبختیان - هادی شیخی - سید صالح اعتمادی

۱. [۱۰] برای آرایه مقابل الگوریتم Merge sort را مرحله به مرحله اجرا کنید و برای هر مرحله تغییرات حاصل در آرایه را نشان دهید.

(12, 6, 9, 7, 12, 15, 3, 10)



۲. [۱۵] با توجه به میحث تقسیم و حل ارائه شده در کلاس، میدانیم که ضرب دو چند جمله ای با درجه n در زمان $O(n^2)$ قابل انجام می باشد. نقش بازنویسی حاصل عبارت $ad + cb$ به صورت $(a + b)(c + d) - ac - bd$ را در بهبود پیچیدگی محاسباتی الگوریتم ضرب چند جمله ای توضیح داده و همچنین رابطه بازگشتی، پیچیدگی محاسباتی و پیچیدگی حافظه ای آنرا با استفاده از Master Theorem به دست بیاورید.



۳. [۱۵] آرایه ای به طول ۵۱۱ داریم . میخواهیم با استفاده از binary search عدد ۱۱۰ را در این آرایه جستجو کنیم . اگر این عدد در آرایه موجود نباشد چه تعداد مقایسه در اجرا الگوریتم رخ داده است ؟
 کدام ایندکس های آرایه مورد مقایسه قرار گرفتند ؟
 در صورتی که الگوریتم اجرایی linear search باشد تعداد مقایسه ها چه تفاوتی می کند ؟

۴. [۲۰] درستی یا نادرستی عبارات زیر را با علامت (✓) یا (X) مشخص کنید. دلیل خود را در نقطه چین زیر هر عبارت توضیح دهید.

- (a) ___ The following array is a max heap: [10, 3, 5, 1, 4, 2]

- (b) ___ In max-heaps, the operations insert, find-max, and find min all take $O(\log n)$ time.

- (c) ___ stack is the most suitable data structures if you only need to implement recursion in a programming language

- (d) ___ tree is the most suitable data structure if you need to store the directory structure on your hard drive

۵. [۱۰] چگونه با استفاده از یک *priopityQueue* یک *stack* پیاده سازی کنیم؟

۶. [۱۵] توضیح دهید قطعه کد روبرو چه کاری انجام میدهد .

```
void function()
Node * current ← head
Node * pre ← Null
Node * next ← Null
while current ≠ NULL do
    next ← current- > next
    current- > next ← pre
    pre ← current
    current ← next
end
head ← pre
```

سپس الگوریتم را بر روی *LinkList* مقابل به صورت مرحله به مرحله اجرا کنید و خروجی را نمایش دهید .

1- > 2- > 3- > null

۷. [۱۵] یک اسب را در صفحه شطرنج $n * n$ در نظر بگیرید . خانه هایی که اسب میتواند به آن ها برود در تصویر زیر قابل مشاهده است . یک الگوریتم برای محاسبه تعداد راه هایی که در آن می توانید با $k \geq 0$ حرکت از خانه (i_s, j_s) به (i_t, j_t) برود طراحی کنید . همچنین پیچیدگی محاسباتی و پیچیدگی حافظه ای آنرا به دست بیاورید .

