



دانشکده علوم ریاضی و آمار



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۷ ساختمان داده‌ها و الگوریتم‌ها

نگارنده: پروین کاظمی

۱۰ مهر ۱۴۰۰

فهرست مطالب

- ۱ مرتب سازی درجی- تحلیل درستی و کارایی
- ۲ الگوریتم‌های درجا و برون از جا
- ۳ الگوریتم مرتب سازی درجی

۱ مرتب سازی درجی- تحلیل درستی و کارایی

در این بخش سعی بر آن است تا نحوه اثبات درستی و تحلیل کارایی الگوریتم‌ها را با استفاده از الگوریتم مرتب‌سازی درجی به عنوان یک الگوریتم منتخب شرح دهیم. برای این منظور آرایه A با محتوای زیر را در نظر بگیرید:

آرایه A: ۸، ۲، ۴، ۹، ۳، ۶، ۳

الگوریتم مرتب‌سازی درجی با فرض داشتن یک آرایه اضافی:

مرحله	محتوای آرایه
مرحله ۱	۸
مرحله ۲	۲، ۸
مرحله ۳	۲، ۴، ۸
مرحله ۴	۲، ۴، ۸، ۹
مرحله ۵	۲، ۳، ۴، ۸، ۹
مرحله ۶	۲، ۳، ۴، ۶، ۸، ۹
مرحله ۷	۲، ۳، ۳، ۴، ۶، ۸، ۹

الگوریتم مرتب‌سازی درجی بدون استفاده از آرایه اضافی:

آرایه A: ۸، ۲، ۴، ۹، ۳، ۶، ۳

مرحله	محتوای آرایه
مرحله ۱	۸، ۲، ۴، ۹، ۳، ۶، ۳
مرحله ۲	۸، ۲، ۴، ۹، ۳، ۶، ۳ ۲، ۸، ۴، ۹، ۳، ۶، ۳
مرحله ۳	۲، ۸، ۴، ۹، ۳، ۶، ۳ ۲، ۴، ۸، ۹، ۳، ۶، ۳
مرحله ۴	۲، ۴، ۸، ۹، ۳، ۶، ۳
مرحله ۵	۲، ۴، ۸، ۹، ۳، ۶، ۳ ۲، ۴، ۸، _، ۹، ۶، ۳ ۲، ۴، _، ۸، ۹، ۶، ۳ ۲، _، ۴، ۸، ۹، ۶، ۳ ۲، ۳، ۴، ۸، ۹، ۶، ۳
مرحله ۶	۲، ۳، ۴، ۸، ۹، ۶، ۳ ۲، ۳، ۴، ۸، _، ۹، ۶، ۳ ۲، ۳، ۴، _، ۸، ۹، ۶، ۳ ۲، ۳، ۴، ۶، ۸، ۹، ۶، ۳
مرحله ۷	۲، ۳، ۴، ۶، ۸، ۹، ۳، ۶ ۲، ۳، ۴، ۶، ۸، _، ۹ ۲، ۳، ۴، ۶، _، ۸، ۹ ۲، ۳، ۴، _، ۶، ۸، ۹ ۲، ۳، ۳، ۴، ۶، ۸، ۹

نکته

در پایان مرحله k -ام، k عنصر ابتدایی آرایه A مرتب شده است.

۲ الگوریتم‌های درجا و برون از جا

الگوریتم‌ها را برحسب ارتباط اندازه حافظه مصرفی با اندازه ورودی می‌توان به دو رده زیر تقسیم‌بندی کرد.

- الگوریتم درجا (in-place): الگوریتمی است که نیاز به فضای اضافی (داده ساختار جدید) برای دستکاری ورودی و تولید خروجی ندارد و از همان فضای ورودی و یک مقدار ثابت حافظه که وابسته به اندازه ورودی نیست، استفاده می‌کند. چنین الگوریتم‌هایی طور خاص برای کاربردهایی که در آنها ظرفیت حافظه محدود است، بیشتر مورد توجه قرار می‌گیرد.
- الگوریتم برون از جا (out-of-place): نیاز به فضای اضافی برای تولید خروجی دارد.

۳ الگوریتم مرتب سازی درجی

شبه کد مربوط به الگوریتم مرتب سازی درجی در ادامه آورده شده است.

Algorithm 1 Insertion-Sort($A[1..n]$)

```
1: for  $k = 2$  to  $n$  do  
2:    $key \leftarrow A[k]$   
3:    $i \leftarrow k - 1$   
4:   while  $i > 0$  and  $A[i] > key$  do  
5:      $A[i + 1] \leftarrow A[i]$   
6:      $i \leftarrow i - 1$   
7:    $A[i + 1] \leftarrow key$ 
```

مفهوم ناوردایی (Invariant): همانطور که از نام آن مشخص است، بیانگر یک خصیصه (ویژگی) است که در طول اجرای یک الگوریتم ثابت است (تغییر نمی کند). این مفهوم ما را قادر می سازد تا اثبات درستی الگوریتم ها را نشان دهیم. آنچه که به طور خاص برای اثبات درستی الگوریتم ها به کار گرفته می شود ناوردایی حلقه (loop invariant) است.