

## نمونه سوالات مربوط به بخش 5.5

1.5.5 برای اعداد زیر مرتب سازی حبابی را دنبال کنید.

50 , 57 , 99 , 34 , 56 , 89

راه حل:

در هر فاز بزرگترین عنصر انتهای لیست مربوطه قرار میگیرد و در هر مرحله لیست کوچکتر حاصل میشود و در نهایت لیست دو عنصره مرتب میشود.

مرحله اول : 50 , 57 , 99 , 34 , 56 , 89 --> 50 , 57 , 34 , 56 , 89 , 99

مرحله دوم : 50 , 57 , 34 , 50 , 89 --> 50 , 34 , 56 , 57 , 89

مرحله سوم : 50 , 34 , 56 , 57 --> 34 , 50 , 56 , 57

مرحله چهارم : 34 , 50 , 56 --> 34 , 50 , 56      لیست نهایی : 34 , 50 , 56 , 57 , 89 , 99

2.5.5 برای اعداد زیر مرتب سازی حبابی را دنبال کنید.

10 , 30 , 50 , 40 , 90 , 20

راه حل:

10	30	50	40	90	20	
10	30	<del>50</del>	<del>40</del>	<del>90</del>	<del>20</del>	مرحله اول
		40	50	20	90	
10	30	40	<del>50</del>	<del>20</del>	90	مرحله دوم
			20	50		
10	30	<del>40</del>	<del>20</del>	50	90	مرحله سوم
		20	40			
10	<del>30</del>	<del>20</del>	40	50	90	مرحله چهارم
	20	30				
10	20	30	40	50	90	مرحله پنجم

:

## نمونه سوالات مربوط به بخش 5.6

1.5.6 برای اعداد زیر مرتب سازی درجی را دنبال کنید..

70 , 57 , 99 , 34 , 56 , 89

راه حل:

نخست عنصر دوم را با عنصر اول مقایسه کرده لیست دو عنصری را مرتب میکنیم سپس عنصر سوم با دو عنصر قبلی تشکیل لیست مرتب سه تایی میدهند و الی آخر.

مرحله اول : 70 , 57 --> 57 , 70

مرحله دوم : 57 , 70 , 99 --> 57 , 70 , 99

مرحله سوم : 57 , 70 , 99 , 34 --> 57 , 70 , 34 , 99 --> 34 , 57 , 70 , 99

مرحله چهارم : 34 , 57 , 70 , 99 , 56 --> 34 , 57 , 70 , 56 , 99 --> 34 , 56 , 57 , 70 , 99

مرحله پنجم : 34 , 56 , 57 , 70 , 99 , 89 --> 34 , 56 , 57 , 70 , 89 , 99

2.5.6 آرایه زیر را به روش درجی مرتب کنید.

4	8	5	2	6
---	---	---	---	---

راه حل:

n	i	j	y	A	1	2	3	4	5
5	2	1	8		4	8	5	2	6
	3	2	5		4	8			
	4	1	2		4	5	8		
		3	6		2	4	5	8	
		2			2	4	5	6	8
		1							
		0							
		4							
		3							

3.5.6 آرایه زیر را به روش درجی مرتب نمایید.

7	8	5	2	4	6	3
---	---	---	---	---	---	---

راه حل:

```

7| 8 5 2 4 6 3
7 8| 5 2 4 6 3
7 5| 8 2 4 6 3
5 7| 8 2 4 6 3
5 7 8| 2 4 6 3
2 5 7 8 | 4 6 3
2 4 5 7 8 | 6 3
2 4 5 6 7 8 | 3
2 3 4 5 6 7 8 |

```

## نمونه سوالات مربوط به بخش 5.7

1.5.7 مرتب سازی  $n$  عدد ذخیره شده در  $A$  را در نظر بگیرید که ابتدا کوچکترین عنصر  $A$  را یافته و آنرا با عنصر  $A[1]$  تعویض میکنیم. سپس دومین عنصر کوچکتر  $A$  را یافته و آنرا با  $A[2]$  تعویض میکنیم. این روند را برای  $n-1$  عنصر اول ادامه میدهیم. شبه کدی برای این الگوریتم که به مرتب سازی انتخابی معروف است بنویسید. چرا لازم است این الگوریتم برای  $n-1$  عنصر اول به جای  $n$  عنصر اجرا شود؟ زمان اجرای الگوریتم مرتب سازی را در بهترین حالت و در بدترین حالت بیان کنید.

راه حل:

$Selection\_Sort(A)$

$n \leftarrow length[A]$

for  $j \leftarrow 1$  to  $n-1$

*do smallest*  $\leftarrow j$

*for*  $l \leftarrow j+1$  *to*  $n$

*do if*  $A[i] < A[\text{smallest}]$

*then smallest*  $\leftarrow i$

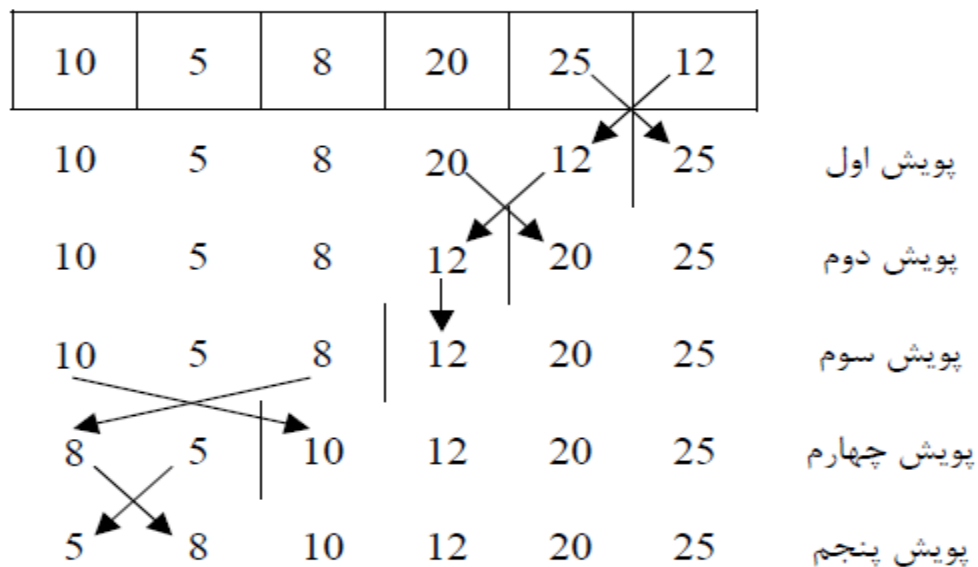
*exchange*  $A[j] \leftrightarrow A[\text{smallest}]$

زیر آرایه  $A[1 \dots j-1]$  شامل  $j-1$  کوچکترین عنصر در آرایه  $A[1 \dots n]$  میباشد و همچنین این آرایه به صورت مرتب شده میباشد.  $n-1$  عنصر اول در زیر آرایه  $A[1 \dots n-1]$  شامل  $n-1$  کوچکترین عنصر مرتب شده میباشد بنابراین عنصر  $A[n]$  بزرگترین عضو این آرایه میباشد. هزینه اجرای این الگوریتم در هر دو حالت مذکور  $n$  به توان 2 میباشد.

2.5.7 آرایه زیر را به روش انتخابی مرتب نمایید.

10	5	8	20	25	12
----	---	---	----	----	----

راه حل:



3.5.7 آرایه زیر را به روش انتخابی مرتب کنید.

2	8	4	1	7
---	---	---	---	---

راه حل:

- 1)  $2 \underline{8} 4 1 \underline{7} \rightarrow 2 7 4 1 8$
- 2)  $2 \underline{7} 4 \underline{1} 8 \rightarrow 2 1 4 7 8$
- 3)  $2 1 \underline{4} 7 8 \rightarrow 2 1 4 7 8$
- 4)  $\underline{2} 1 4 7 8 \rightarrow 1 2 4 7 8$

در مرحله ی اول، کل لیست از ابتدا تا انتها بررسی شده و بزرگترین عنصر با عنصر انتهای لیست نامرتب جابجا می شود.

در مرحله ی دوم، پیمایش از ابتدای لیست تا عنصر چهارم صورت گرفته و بزرگترین عنصر با عنصر انتهای آن جابجا می شود.

علت این که چرا عنصر پنجم بررسی نمی شود کاملاً مشخص است. این عنصر در مرحله ی قبل به عنوان بزرگترین عنصر به انتهای

لیست منتقل شده است و به طور حتم نیاز به جا به جایی ندارد.

در مرحله ی سوم، عناصر اول تا سوم بررسی شده و بزرگترین عنصر به انتهای آن منتقل می شود:

و در مرحله ی آخر دو عنصر باقیمانده مقایسه می شوند:

و به این ترتیب لیست مرتب می شود.

---

## نمونه سوالات مربوط به بخش 5.8

5.8.1 K تا آرایه مرتب شده از اعداد صحیح به شما داده شده است. این K آرایه در مجموع شامل n عدد هستند می خواهیم این آرایه ها را ادغام و آرایه ای مرتب شامل د عدد به دست بیاوریم. الگوریتمی از  $O(n \log k)$  آرایه دهید که این کار را انجام دهد.

راه حل:

ابتدا یک آرایه به طول n در نظر میگیریم. سپس عناصر k لیست را عنصر به عنصر با یکدیگر مقایسه نموده و کوچکترین عضو را از بین k آرایه پیدا میکنیم (به عنوان مثال عنصر i) در آرایه مذکور میریزیم و عنصر i را از آرایه ای که شامل i میباشد حذف میکنیم و تمامی عناصر آرایه ای را که حاوی عنصر i بود را یک واحد شیفت میدهیم. و این کار را همچنان ادامه میدهیم تا تمامی k آرایه خالی شود.

2.5.8 آرایه زیر را به روش ادغامی مرتب نمایید.

5	1	7	2
---	---	---	---

راه حل:

5	1	7	2
5	1	7	2
5	1	7	2
1	5	2	7
1	2	5	7

L = 3	U = 3	L = 4	U = 4
L = 3	U = 3	i = 3	
L = 1	U = 1	L = 2	U = 2
L = 1	U = 2	i = 1	
L = 1	U = 4	i = 2	

3.5.8 آرایه زیر را به روش ادغامی مرتب نمایید.

5	3	1	4
---	---	---	---

راه حل::

	1	2	3	4
A	5	3	1	4
	Merge sort ( 1 , 4 )			
	1	2	3	4
	3	5	1	4
	1	3	4	5

L = 4	U = 4
L = 3	U = 4
L = 3	U = 4    i = 3
L = 2	U = 2
L = 1	U = 1
L = 1	U = 2    i = 1
L = 1	U = 4    i = 2

4.5.8 جدول زیر را کامل نمایید.

نام الگوریتم	بهترین حالت	حالت متوسط	بدترین حالت	ویژگی ها
مرتب سازی حبابی				
مرتب سازی درجی				
مرتب سازی انتخابی				
مرتب سازی ادغامی				

راه حل:

نام الگوریتم	بهترین حالت	حالت متوسط	بدترین حالت	ویژگی ها
مرتب سازی حبابی	$O(n)$	$O(n^2)$	$O(n^2)$	پایدار است و درجا
مرتب سازی درجی	$O(n)$	$O(n^2)$	$O(n^2)$	پایدار است و درجا
مرتب سازی انتخابی	$O(n^2)$	$O(n^2)$	$O(n^2)$	پایدار نیست و درجا
مرتب سازی ادغامی	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	پایدار است و غیر درجا