

بسم الله الرحمن الرحيم

Subject ۹۸۴۴۴۴
Date

تکلیف ۳ از خان داد

سوال الف) در حالت تصادفی یا هر دو موقعیت average case worst case Best case

ما مقایسه می کنیم که البته ممکن حالت average در اینجا کافی بود

	Best	average	worst
Bubble sort	n^2	n^2	n^2
Insertion sort	n^2	n^2	n^2
Selection sort	n^2	n^2	n^2
Merge sort	$n(\log n)$	$n(\log n)$	$n(\log n)$
Quick sort	$n(\log n)$	$n(\log n)$	n^2
Heap sort	$n(\log n)$	$n(\log n)$	$n(\log n)$

در Best case برای bubble sort زمانی است که عناصر sort شده به همین گونه است. در average case

هم برای $n \log n$ عناصر نه کامل sort و نه کامل sort برعکس اند.

وقتی می گوید از قبل sort شده یعنی Best case. با توجه به جدول بالا در Bubble sort

Insertion sort $O(n)$ است که بهینه ترین.

[illegible]

اما اگر جای بیغ ندانم مقدار ادب آن که True بودی مانند پس بداز کجا؟ حکیم اگر True بود

- subtracks (if any)

الف) $\log \text{ Pivot} = 2$ $\Rightarrow \log 109293$ و $\log 293930 \Rightarrow 0, \log 109293 + 293930 \log 109293$

$$= 10 \cdot 10^2 \cdot 10^3 \cdot 10^4 \cdot 10^5 \cdot 10^6 \cdot 10^7 \cdot 10^8 \cdot 10^9 \cdot 10^{10} = 10^{55}$$
$$\Rightarrow \frac{0.9}{1} = \frac{0.9 \log 10^7}{\frac{1}{10}} + 0.9 \log 10^7 \Rightarrow \frac{0.9 \times 7}{1} = \frac{0.9}{10} + 0.9 \log 10^7$$
[illegible][illegible]

$\Rightarrow 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 362880$

$$\text{Ans: } P_1 v + c^2 \rightarrow 0, \partial_0 \gamma^0 \partial_0 \gamma^0 \Rightarrow \underset{0}{\partial_0} \partial_0 \gamma^0 \partial_0 \gamma^0 \Rightarrow \underset{0}{\partial_0} \partial_0 \gamma^0 \partial_0 \gamma^0 \Rightarrow \partial_0 \partial_0 \gamma^0 \partial_0 \gamma^0$$
$$\Rightarrow \underline{0, 5, 0, 0, 1} \Rightarrow \underline{0, 1, 0, 0, 0} \Rightarrow \underline{0, 0, 1, 0, 0}$$

$\text{pivot} = 5 \quad 09 \rightarrow 09 \rightarrow 29 \rightarrow 09$

Ans: pivot = 0
 $10 \Rightarrow 20 \Rightarrow 30$

برای نیکی را = هم دقیقاً مانند یک $\rightarrow \{ \log 1, \log 1 \} \rightarrow \text{Pivot} = 9$ $\frac{9}{2}$

$\Rightarrow 092, 29, 09, 09, 9, 1, 09, 1, 09, 1$



PARS GARMA

تاریخ:

موضوع:

عنوان زارع

بخش ۲.۱

به عنصر عدد اول و آخر و عدد وسط را مقایسه می‌کنیم و
و عنصر میانی در مقایسه را به عنوان Pivot در نظر گرفته و
مقایسه را انجام می‌دهیم مثلاً:

۳	۷	۶	۵	۴	۱۳	۹
۲			۲			۱

پس ۳ و ۹ و ۵ و ۴ به عنوان

Pivot در نظر گرفته می‌شود.

بخش ۲.۲ (ح)

برای اعدادی که در یک محدوده اند و تکراری شوند
بهترین الگوریتم counting sort است که به شکل زیر است

counting sort (mat & k)

for i = 1 to k

P[i] = 0

for j = 1 to n

P[m[j]]++

for i = 2 to k

P[i] += P[i-1]

for d = n to 1 (کمری و جدولی)

f[P[m[j]]] = m[j]

P[m[j]]--

m

۵	۹	۳	۱۰	۵	۳	۵	۱۰	۱۱	۱۰	۱۱	۱۰	۵
---	---	---	----	---	---	---	----	----	----	----	----	---

P

۳	۱	۲	۳	۲	۲	۲	۲	۲	۲	۲	۲	۲
---	---	---	---	---	---	---	---	---	---	---	---	---

۱۰	۷	۶	۶	۶	۶	۶	۶	۶	۶	۶	۶	۶
----	---	---	---	---	---	---	---	---	---	---	---	---

f

۵	۱۰	۱۰	۹	۵	۵	۵	۵	۵	۵	۵	۵	۵
---	----	----	---	---	---	---	---	---	---	---	---	---

^[0] { 1300, 204, 1500, 209, 1204, 1120, 1345, 0814, 1398, 0201 }
^[1] 1399, 1229, 1345, 0814, 1345, 0209, 1345, 0209, 1345, 0209, 1345, 0209

[2] 13

merge sort(arr, left, right)

if (left < right)

{

mid = floor((left + right) / 2);

merge sort(arr, left, mid)

merge sort(arr, mid + 1, right)

merge(arr, left, mid, right)

Ex:

1300, 204	1500, 209	1204, 1120	1345, 0814	1398, 0201
-----------	-----------	------------	------------	------------

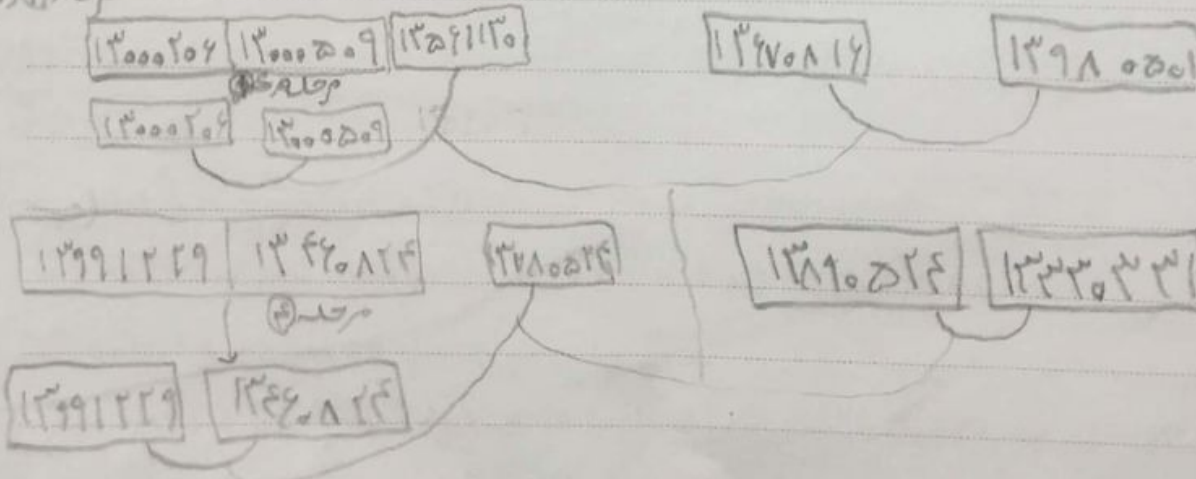
1399, 1229	1345, 0814	1345, 0209	1345, 0209	1345, 0209
------------	------------	------------	------------	------------

Ex:

1300, 204	1500, 209	1204, 1120	1345, 0814	1398, 0201
-----------	-----------	------------	------------	------------

1399, 1229	1345, 0814	1345, 0209	1345, 0209	1345, 0209
------------	------------	------------	------------	------------

Ex, Ex:



بعد از divide کردن و با توجه به فرمول و سود و زیان نه شده یعنی رفته و در نهایت
(در هر مرحله دو عمل شده در مرحله اول)

به مرحله merge کردن می آیم که در این مرحله عناصر که به هم متناسب می شوند و یک آرایه ی سازنده می آید

ما با هم متناسب می شوند و به این گونه که روی هر آرایه یک اشاره گر می گذاریم و با هم متناسب با اشاره گر آرایه دیگر می گردیم

merge

در آرایه جدید قرار می گیرد و اشاره گر یک جلوی ر

مرحله ۱ و ۲

۱۳۰۰۰۲۰۶ ۱۳۰۰۰۵۰۹ ۱۳۵۶۱۱۳۰

۱۳۴۷۰۸۱۶ ۱۳۹۸۰۵۰۱

★

۱۳۴۶۰۸۱۴ ۱۳۹۹۱۲۲۹ ۱۳۷۸۰۵۲۴

۱۳۳۳۰۳۳۱ ۱۳۸۹۰۵۲۴

★★

متناسب

مرحله ۳

۱۳۰۰۰۲۰۶ ۱۳۰۰۰۵۰۹ ۱۳۵۶۱۱۳۰

۱۳۴۷۰۸۱۶ ۱۳۹۸۰۵۰۱

★★

۱۳۴۶۰۸۱۴ ۱۳۷۸۰۵۲۴ ۱۳۹۹۱۲۲۹

۱۳۳۳۰۳۳۱ ۱۳۸۹۰۵۲۴

مرحله ۳

متناسب

۱۳۰۰۰۲۰۶ ۱۳۰۰۰۵۰۹ ۱۳۵۶۱۱۳۰ ۱۳۴۷۰۸۱۶ ۱۳۹۸۰۵۰۱

متناسب

۱۳۳۳۰۳۳۱ ۱۳۴۶۰۸۱۴ ۱۳۷۸۰۵۲۴ ۱۳۸۹۰۵۲۴ ۱۳۹۹۱۲۲۹

مرحله ۴

۱۳۰۰۰۲۰۶ ۱۳۰۰۰۵۰۹ ۱۳۳۳۰۳۳۱ ۱۳۴۶۰۸۱۴ ۱۳۵۶۱۱۳۰ ۱۳۷۸۰۵۲۴ ۱۳۸۹۰۵۲۴ ۱۳۹۸۰۵۰۱ ۱۳۹۹۱۲۲۹

[illegible]

ادامه سوال 3
Radix Sort Complexity

سورت n عدد برای رقم در رنج 1 تا $k \leftarrow o(n \cdot k)$ رقم هر عدد رقم داشته باشد

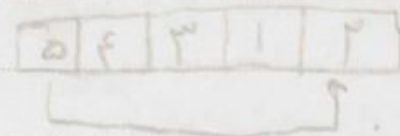
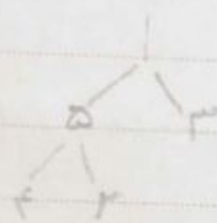
$o(n \cdot k)$

بناچار به آنکه تعداد ارقام هر کدام k تا و تعداد اعداد است o

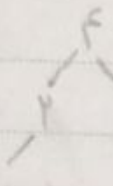
$$k=10 \quad a=9 \Rightarrow o(9n + 9)$$

گراف و آرایه ها

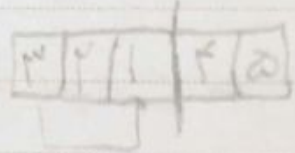
کلاس



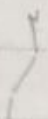
حذف 5 از
ریشه



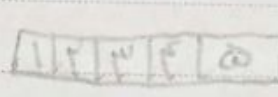
حذف 4 از
ریشه



حذف 3 از
ریشه



حذف 2 از
ریشه



روش Heap Sort ابتدا عنصر max را در Root قرار می دهیم سپس یک بار Reheapify می کنیم بطل

دیگر در یک Loop می گذاریم و تا زمانی که Index عنصر سورت از آخر به خانه اول برسد ادامه می یابد

در Loop، عنصر Root جایگزین را با آخرین عنصر sort شده جای می شود و از Tree

حذف شده و سپس Reheapify می کنیم.

روش دومین گونه است که عناصر را ابتدا با merge sort اولیه کرده. سپس روی هر ایندکس از ابتدای

آرایه شروع به مرتب کردن و قدر مطلق هر عنصر با عنصر طی شد 65 به مقایسه با K می پردازیم. پس وقتی

یافته شد به تغییر به نام Counter را یک واحد زیاد می کنیم و باز در Continue

یک واحد در آرایه جلوی دوم (با فرض تکراری نداشته و ورودی که اگر باشد بعد از

اینکه قدر مطلق برابر نبود در ادین جا Continue می کنیم. در نهایت مقدار نهایی را چاپ می کنیم