

موضوع: مرتب سازی داخلی - ساله انتخاب ✓

* مرتب سازی شمارشی: $m \leq n$ - n تا عدد: $O(m+n)$

* مرتب سازی سطلی bucket sort

* مرتب سازی مبنایی Radix sort

مرتب سازی سطلی: ایده اصلی: عناصر را با استفاده از یک تابع هاش مرتب داخل تعدادی سطل قرار می دهیم. بعد عناصر داخل هر سطل را مرتب و سپس عناصر مرتب شده سطل های مختلف را پشت سر قرار می دهیم

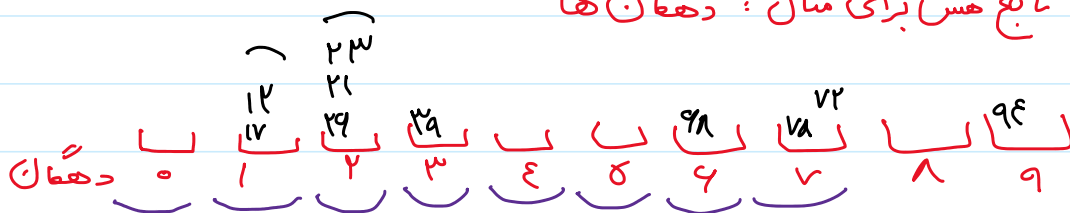
تابع هاش مرتب: $x < y \Rightarrow f(x) < f(y)$ مرتب است

مثال: ۷۸ - ۱۷ - ۳۹ - ۲۶ - ۷۲ - ۹۴ - ۲۱ - ۱۲ - ۲۳ - ۶۸

۲۶ ۱۷

تابع درهم سازی مرتب؟

تابع هاش برای مثال: دهگان ها



۱۲ - ۱۷ - ۲۱ - ۲۳ - ۲۶ - ۳۹ - ۶۸ - ۷۲ - ۷۸ - ۹۴

* داخل هر سطل: یک مرتب سازی دلخواه.

insertion sort

زمان اجرا در بدترین حالت: هم داخل یک سطل بودند: $O(n^2)$

در حالت متوسط: با فرض کیندازی اگر m تا سطل داشته باشیم،

داخل هر سطل $\frac{n}{m}$ عنصر قرار می گیرد.

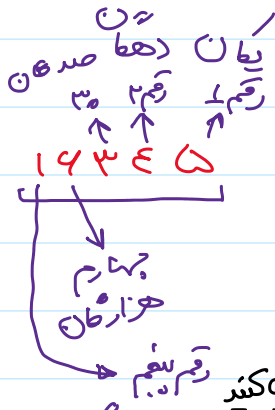
در حالت متوسط : با فرض اینکه m و n متناسب باشند :

داخل هر سطل $\frac{n}{m}$ عنصر قرار می‌گیرد.

$$O(n) + O\left(\frac{n^2}{m^2} \times m\right) = O\left(\frac{n^2}{m} + n\right) *$$

$$? = m$$

* مرتب سازی مبتنی بر Radix sort



فرض کنید کلیدها ورودی اعداد حداقل d رقمی هستند.

$$d = 5$$

الگوریتم :

$$for(i: 1 \rightarrow d) \{$$

عنصر آرایه A را بر حسب رقم i ام با استفاده از یک مرتب سازی پایدار

↓
counting sort
}

$$\frac{0-9}{\uparrow}$$

کل زمان الگوریتم: $O(nd)$

۱۲۳۴ - ۱۲۳۶ - ۴۱۹۱ - ۴۲۱۰ - ۱۹۴۰ - ۲۳۲۶ - ۹۱۰۱

↓
۴۲۱۰ - ۱۹۴۰ - ۴۱۹۱ - ۹۱۰۱ - ۱۲۳۴ - ۱۲۳۶ - ۲۳۲۶

↓
۹۱۰۱ - ۴۲۱۰ - ۲۳۲۶ - ۱۲۳۴ - ۱۲۳۶ - ۱۹۴۰ - ۴۱۹۱

↓
۹۱۰۱ - ۴۱۹۱ - ۴۲۱۰ - ۱۲۳۴ - ۱۲۳۶ - ۲۳۲۶ - ۱۹۴۰

↓
✓ ۱۲۳۴ - ۱۲۳۶ - ۱۹۴۰ - ۲۳۲۶ - ۴۱۹۱ - ۴۲۱۰ - ۹۱۰۱

پایه مرتب سازی اصلی

مسئله انتخاب : آرایه A شامل n عدد داده شده است. عنصر

مرتبه k ام آرایه را انتخاب کنید [عنصری که در حالت مرتب شده

مسئله انتخاب : آرایه A شامل n عدد داده شده است. عنصر مرتبه K ام آرایه را انتخاب کنید [عنصری که در حالت مرتبه شده در جایگاه K ام قرار می گیرد]

حالت خاص :

$O(n)$ min عنصر : $K=1$

$O(n)$ max " : $K=n$

$O(n)$: $K=2$

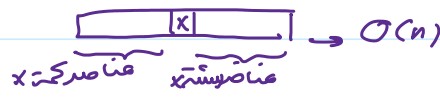
$O(n \log n) \leftarrow$ مرتبه سازی ؟ : $K=\frac{n}{2}$ عنصر میانه

*** $O(n)$

روش $K \log K$: استفاده از Heap : $K=\frac{n}{2}$ $\leftarrow O(n \log n)$

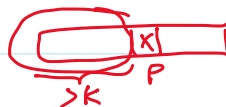
روش پیشنهادی : (الگوریتم q-Select) : عنصر K ام.

$O(n)$ ۱- آرایه را بر حسب عنصر دلخواه x پارتیشن کنید. [P محل x باشد]



۲- اگر $P=K$: x را برگردانیم

← ۳- اگر $P > K$: به طور بازگشتی عنصر K ام را در $A[1 \dots P-1]$ پیدا کنیم

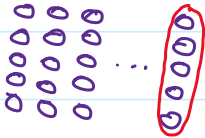


۴- اگر $P < K$: به طور بازگشتی عنصر $K-P$ ام آرایه $A[P+1 \dots n]$ را پیدا کنیم.

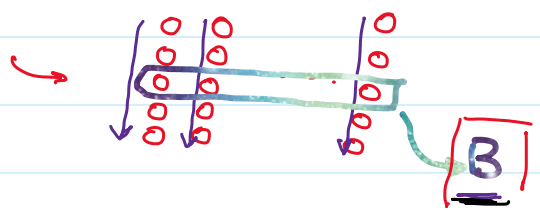
زمان اجرا در بدترین حالت : [هنگامی که عنصر آخر را برگردانیم] $O(n^2)$

حالت متوسط : $O(n)$

الگوریتم f-select : $T(n)$



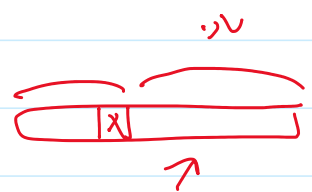
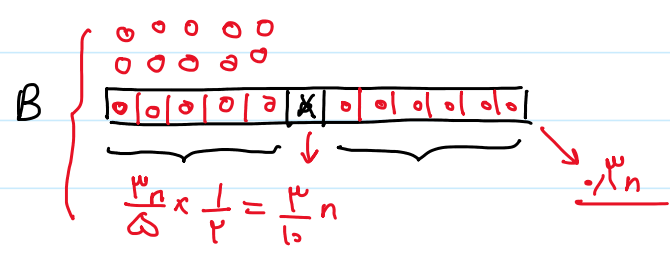
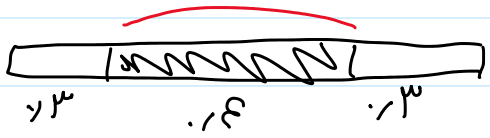
۱. $O(n)$ → عناصر A را به $\frac{n}{5}$ دسته با اندازه ۵ تقسیم کنید.
 ۲. $O(n)$ → تمام دسته‌های ۵ تایی را مرتب کنید.
 ۳. $O(n)$ → آرایه B شامل تمام عناصر میانی هر دسته را تسلیل دهید.
 ۴. $T(\frac{n}{5})$ → عنصر میانه B را با استفاده از $fselect$ به طور بازگشتی پیدا کنید.
- فرض کنید x عنصر میانه B است.



۵. $O(n)$ → A را بر حسب عنصر x ، پارس‌ترین می‌کنیم [P محل x است]
 $P = K$: x را برمی‌گرداند
 $P < K$: عنصر $K-P$ ام را در آرایه $A[P+1 \dots n]$ می‌کنیم با $fselect$
 $P > K$: عنصر K ام را در آرایه $A[1 \dots P-1]$ پیدا می‌کنیم با $fselect$

س
۱

* قضیه x [عنصر میانه B] حداقل از $\frac{3}{10}n$ عناصر کوچکتر و حداقل از $\frac{3}{10}n$ عناصر بزرگتر است.



$$T(n) = T(\frac{3}{10}n) + T(\frac{3}{10}n) + O(n)$$

$T(1) = C_1$ \downarrow
 $C_2 n$

$$T(n) \leq T(n/2) + T(n/2) + C_2 n$$

$$T(1) \leq C_1$$

حکم: $T(n) \leq 2 \cdot (C_1 + C_2) n$

استقرا: $\forall n$ برای همه از n درست است

$$T(n) \leq \underbrace{T(n/2)} + \underbrace{T(n/2)} + C_2 n$$

$$\leq \underbrace{2 \cdot (C_1 + C_2) \cdot n/2} + C_2 n$$

$$= \underbrace{2(C_1 + C_2)n/2 + C_2 n}$$

$$= 1 \cdot (C_1 + C_2) n + C_2 n$$

$$\leq 2 \cdot (C_1 + C_2) n \quad \square$$

$$= O(n)$$