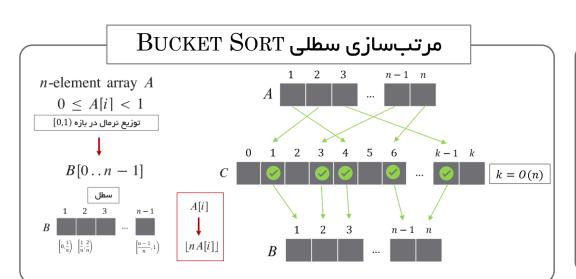


مرور جلسه قبل



RADIX SORT مرتبسازی خطی

اطلاعات مازاد: اعداد دارای d رقم و هر رقم k مقدار متفاوت

l					1
329	720	720	329	RADIX-SORT (A, d)	
457	355	329	355	1 for $i = 1$ to d	
657	436	436	436	2 use a stable sort to sort array A on digit i	
839	457յր.	839 mijii-	457	,	
436	657	355	657		
720	329	457	720		
355	839	657	839	Q(d(n+k)) + -1 + 1 = 1	
I				$H(d(n \perp k)) \cdot \cdot \cdot \cdot \cdot \cdot $	4

اگر $\theta(n)$ radix sort باشد زمان اجرای d=O(1) خواهد بود d=0

BUCKET SORT تحلیل زمانی

 $T(n) = \Theta(n) + \sum O(n_i^2)$

 B_i متغیر تصادفی نشانگر تعداد المانها در سطل n_i

 $= \Theta(n) + \sum_{n=1}^{n-1} O\left(2 - \frac{1}{n}\right) = \Theta(n)$

BUCKET-SORT(A)

- 1 let B[0..n-1] be a new array
- $2 \quad n = A.length$
- 3 **for** i = 0 **to** n 1
- 4 make B[i] an empty list
- 5 **for** i = 1 **to** n
- 6 insert A[i] into list $B[\lfloor nA[i] \rfloor]$
- 7 **for** i = 0 **to** n 1
- 8 sort list B[i] with insertion sort
- concatenate the lists $B[0], B[1], \dots, B[n-1]$ together in order

RADIX SORT تحلیل زمانی

اگر n عدد b بیتی داشته باشیم، برای هر عدد مثبت دلخواه $t \leq b$ خواهیم داشت:

رمان اجرایی $\operatorname{radix} \operatorname{sort}$ برای این اعداد $\operatorname{radix} \operatorname{sort}$ خواهد بود

به شرطی که مرتبسازی یایدار استفاده شده heta(n+k) باشد

برای اعداد n و b تعیین $r \leq b$ بگونهای که زمان اجرای $(b/r)(n+2^r)$ را کمینه کند

 $\Theta(n)$:خواهیم داشت $b < \lfloor \lg n \rfloor$ خواهیم داشت

 $\Theta(bn/\lg n)$ حالت دوم: $r = \lfloor \lg n \rfloor$ برای $b \geq \lfloor \lg n \rfloor$ خواهیم داشت:

O(n) بر ابر radix sort زمان اجری $r = \lfloor \lg n \rfloor$ بر ابر با باشد با انتخاب $b = O(\lg n)$ در صورتی که

فصل ۹ کتاب



• میانه و مرتبههای آماری

- کمینه و بیشینه
- امید زمان اجرای الگوریتم انتخاب
- بدترین زمان اجرای الگوریتم انتخاب

9 Medians and Order Statistics 213

- 9.1 Minimum and maximum 214
- 9.2 Selection in expected linear time 215
- 9.3 Selection in worst-case linear time 220

مرتبههای آماری



- انتخاب i امین کوچکترین عنصر در یک آرایه عددی n عنصری \cdot
 - کمینه اعداد: اولین مرتبه آماری
 - بیشینه اعداد: n امین مرتبه آماری \cdot
 - میانه اعداد: عدد وسطی آرایه
 - ام (n+1)/2 اگر n فرد باشد: عنصر n
 - $\frac{n}{2}+1$ و n/2 و n/2 اگر n زوج باشد: عنصرهای •
 - $\lceil (n+1)/2 \rceil$ و میانه بالا $\lceil (n+1)/2 \rceil$ و میانه بالا $\lceil (n+1)/2 \rceil$
 - میانه در این درس برای سادگی میانه پایین فرض میشود
- اعداد آرایه یکتا هستند: هیچ دو عدد تکراری نداریم (صرفا برای راحتی، قابل تعمیم به حالت کلی)

مسئلہ انتخاب Selection Problem



Input: A set A of n (distinct) numbers and an integer i, with $1 \le i \le n$.

Output: The element $x \in A$ that is larger than exactly i-1 other elements of A.

 $1 \leq i \leq n$ ورودی: مجموعه A با n عدد یکتا و یک عدد طبیعی i به شرط i

بزرگتر باشد $x\in A$ بخروجی: عنصر $x\in A$ بزرگتر باشد i-1 نصر دیگر مجموعه $x\in A$ بزرگتر باشد

O(1) راحل حل $O(n \mathrm{lg} n)$: مرتبسازی آرایه با $O(n \mathrm{lg} n)$ و انتخاب i امین عنصر آرایه با $O(n \mathrm{lg} n)$

• هدف: ارائه الگوريتم سريعتر

انتخاب کمینه یا بیشینه

- دانشگاه صنعتی امیر کبیر ریلی تکنیک توران)
- چه تعداد مقایسه برای تعیین کمینه یا بیشینه مورد نیاز است؟
- حد بالا: شروع از اولین عنصر و انجام n-1 مقایسه و نگهداری کمینه یا بیشینه \cdot

MINIMUM(A)

```
1 min = A[1]

2 for i = 2 to A.length

3 if min > A[i]

4 min = A[i]

5 return min
```

- آیا این بهترین کار است؟
- حواب: بله! میتوان نشان داد که n-1 مقایسه حد پایین هم هست \cdot



• تورنومنت اعداد: هرکسی حداقل یک باخت!

انتخاب کمینه و بیشینه بصورت همزمان



• در برخی موارد نیاز به پیدا کردن همزمان ماکزیمم و مینیمم داریم

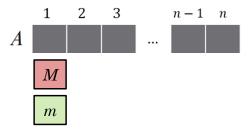
Normalization Formula $X_{normalized} = \frac{(X - X_{minimum})}{(X_{maximum} - X_{minimum})}$

• مثال: نرمال کردن مجموعه ای از اعداد

asymptotically optimal



جواب $\Theta(n)$: تعیین مستقل مینیمم و ماکزیمم هرکدام n-1 مقایسه، مجموعا n-2 مقایسه $\Theta(n)$

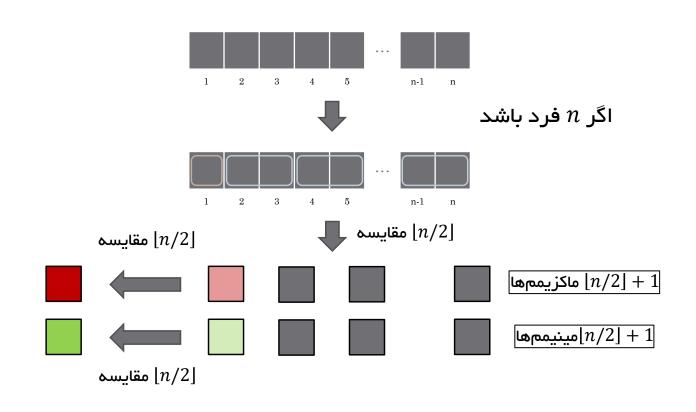


- و روش بهتر؟ جواب: بله! \longrightarrow تعیین همزمان مینیمم و ماکزیمم با حداکثر $3\lfloor n/2 \rfloor$ مقایسه \bullet
 - به جای مقایسه هر عنصر با مینیمم و ماکزیمم فعلی ── مقایسه عناصر بصورت جفت

انتخاب كمينه و بيشينه بصورت همزمان

دانشگاه صنعتی امیر کبیر راس تکنیک تنه إن

• به جای مقایسه هر عنصر با مینیمم و ماکزیمم فعلی → مقایسه عناصر بصورت جفت

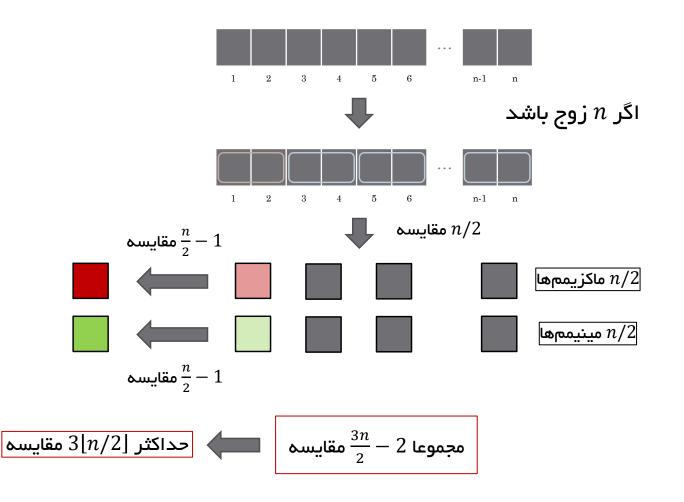


مجموعا $\lfloor n/2 \rfloor$ مقايسه

انتخاب کمینه و بیشینه بصورت همزمان

دانشگاه صنعتی امیر کبیر (بلی تکنیک نیراز)

• به جای مقایسه هر عنصر با مینیمم و ماکزیمم فعلی → مقایسه عناصر بصورت جفت

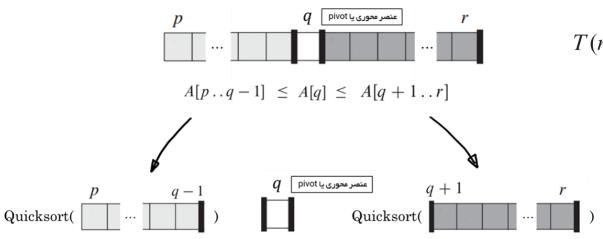


مسئله انتخاب با زمان اجرای متوسط خطی

 $1 \leq i \leq n$ ورودی: مجموعه A با n عدد یکتا و یک عدد طبیعی i به شرط i

خروجی: عنصر $x\in A$ بزرگتر باشد i-1 عنصر دیگر مجموعه $x\in A$ بزرگتر باشد $x\in A$

راه حل با زمان متوسط $\Theta(n)$: روش تقسیم و حل مشابه ایده اصلی مرتبسازی سریع $oldsymbol{\cdot}$



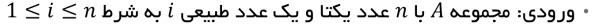
$$T(n) = (T(q) + T(n-q-1)) + \Theta(n)$$

Expected Running Time: $\Theta(n \lg n)$



Expected Running Time: $\Theta(n)$

مسئله انتخاب با زمان اجرای متوسط خطی



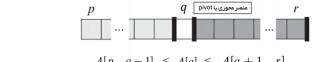
خروجی: عنصر $x\in A$ به شرطی که x دقیقا از i-1 عنصر دیگر مجموعه $x\in A$ بزرگتر باشد $x\in A$

RANDOMIZED-SELECT (A, p, r, i)

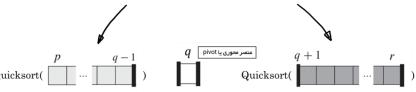
- 1 if p == r
- 2 return A[p]
- 3 q = RANDOMIZED-PARTITION(A, p, r)
- $4 \quad k = q p + 1$
- 5 **if** i == k // the pivot value is the answer
- 6 return A[q]
- 7 elseif i < k
- 8 return RANDOMIZED-SELECT(A, p, q 1, i)
- 9 **else return** RANDOMIZED-SELECT(A, q + 1, r, i k)

RANDOMIZED-PARTITION (A, p, r)

- i = RANDOM(p, r)
- 2 exchange A[r] with A[i]
- 3 **return** PARTITION(A, p, r)



 $A[p \dots q - 1] \leq A[q] \leq A[q + 1 \dots r]$



دارای *q* عنصر

دارای n-q-1 عنصر

$$K = q - p - 1 \le i$$

در نیمه دوم دنبال $i ext{-}k$ امین میگردیم

$$K = q - p - 1 > i$$

در نیمه اول دنبال i امین میگردیم

$$K = q - p - 1 = i$$

برابر i امین عنصر است pivot





- $\Theta(n^2)$:بدترین زمان اجرا حتی برای پیدا کردن مینیمم \bullet
 - $\Theta(n)$:رمان اجر ای متوسط \bullet
- T(n) یک متغیر تصادفی: A[p..r] و زمان اجرای تعیین أ امین عنصر از آرایه A[p..r] یک متغیر تصادفی:
 - $\mathrm{E}[T(n)]$ هدف: محاسبه \bullet
 - استفاده از Randomized-partition:
- 1/n احتمال اینکه آرایه k برای همه مقادیر k در محدوده $k \leq n$ دارای k عنصر باشد برابر با k

 $X_k = I \{ \text{the subarray } A[p ... q] \text{ has exactly } k \text{ elements} \}$

$$\mathrm{E}\left[X_k\right] = 1/n$$

آمار و احتمالات

$$I(A) = egin{cases} 1, & ext{if A happen} \ 0, & ext{if A not happen} \end{cases}$$
 متغیر تصادفی شاخص

درس طراحی الگوریتم(ترم اول ۱ ۰ ۱ ۱) INTRODUCTION TO ALGORITHM

تحلیل زمانی RANDOMIZED-SELECT



 $k \le i$

در نیمه دوم دنبال $i ext{-}k$ امین میگردیم

k > i

در نیمه اول دنبال i امین میگردیم

k = i

برابر i امین عنصر است pivot

تکرار برای n-k عنصر

تکرار برای k-1 عنصر

تمام!

$$T(n) \leq \sum_{k=1}^{n} X_k \cdot (T(\max(k-1, n-k)) + O(n))$$

$$= \sum_{k=1}^{n} X_k \cdot T(\max(k-1, n-k)) + O(n).$$



E[T(n)]

$$\leq \mathbb{E}\left[\sum_{k=1}^{n} X_k \cdot T(\max(k-1, n-k)) + O(n)\right]$$

$$= \sum_{k=1}^{n} E[X_k \cdot T(\max(k-1, n-k))] + O(n)$$

$$= \sum_{k=1}^{n} \operatorname{E}[X_{k}] \cdot \operatorname{E}[T(\max(k-1, n-k))] + O(n)$$

$$= \sum_{k=1}^{n} \frac{1}{n} \cdot E[T(\max(k-1, n-k))] + O(n)$$

امیدریاضی و برخی خواص آن $E(X)=\sum_{i=1}^n p_i x_i$ x_i تعریف امیدریاضی برای متغیر تصادفی مستقل E(aX+b)=aE(X)+b ویژگی خطی بودن امیدریاضی E(XY)=E(X)E(Y) برای متغیر تصادفی مستقل X و Y داریم:

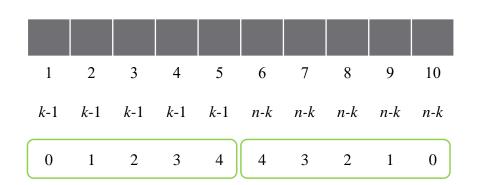


• ادامه محاسبات:

$$E[T(n)] = \sum_{k=1}^{n} \frac{1}{n} \cdot E[T(\max(k-1, n-k))] + O(n)$$

$$\max(k-1, n-k) = \begin{cases} k-1 & \text{if } k > \lceil n/2 \rceil, \\ n-k & \text{if } k \le \lceil n/2 \rceil. \end{cases}$$

$$\mathrm{E}\left[T(n)\right] \le \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} \mathrm{E}\left[T(k)\right] + O(n)$$



$$\mathrm{E}\left[T(n)
ight] \leq rac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} \mathrm{E}\left[T(k)
ight] + O(n) \longrightarrow \mathrm{E}\left[T(n)
ight] = O(n)$$
 جايگذاری

$$E[T(n)] \leq cn$$

حکم

$$E[T(n)] \leq \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} ck + an$$

$$= \frac{2c}{n} \left(\sum_{k=1}^{n-1} k - \sum_{k=1}^{\lfloor n/2 \rfloor - 1} k \right) + an$$

برای nهای کوچکتر از مقداری

$$T(n) = O(1)$$



INTRODUCTION TO ALGORITHM | نرس طراحی الگوریتم (ترم اول ۱۰۶۱) |

تحلیل زمانی RANDOMIZED-SELECT



$$E[T(n)] = \frac{2c}{n} \left(\sum_{k=1}^{n-1} k - \sum_{k=1}^{\lfloor n/2 \rfloor - 1} k \right) + an$$

$$E[T(n)] \leq cn$$

$$= \frac{2c}{n} \left(\frac{(n-1)n}{2} - \frac{(\lfloor n/2 \rfloor - 1) \lfloor n/2 \rfloor}{2} \right) + an$$

$$\leq \frac{2c}{n} \left(\frac{(n-1)n}{2} - \frac{(n/2-2)(n/2-1)}{2} \right) + an$$

$$= c \left(\frac{3n}{4} + \frac{1}{2} - \frac{2}{n} \right) + an$$

$$\leq \frac{3cn}{4} + \frac{c}{2} + an$$

$$= cn - \left(\frac{cn}{4} - \frac{c}{2} - an\right)$$

$$n \ge \frac{c/2}{c/4 - a} = \frac{2c}{c - 4a}$$



$$T(n) = O(1)$$
 for $n < 2c/(c-4a)$



$$E[T(n)] = O(n)$$

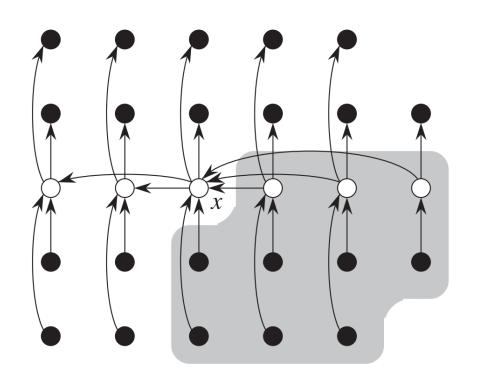
مسئله انتخاب با بدترین زمان اجرای خطی

- دانشگاه صنعتی امیر کبیر دانشگاه صنعتی امیر کبیر (طر تکنیک تعداد)
- $1 \leq i \leq n$ ورودی: مجموعه A با A عدد یکتا و یک عدد طبیعی \cdot
- خروجی: عنصر $x\in A$ بزرگتر باشد i-1 عنصر دیگر مجموعه $x\in A$ بزرگتر باشد
- ایده اصلی: مشابه روش قبل (recursive portioning) اما با ضمانت تقسیم بندی خوب!

- مراحل تابع SELECT
- مرحله ۱: تقسیم ورودی به $\lfloor n/5
 floor$ گروه متشکل از ۵ عنصر و حداکثر یک گروه مابقی اعداد $\lfloor n/5
 floor$
 - مرحله ۲: یافتن میانه در هر $\lfloor n/5 \rfloor$ گروه مرتبسازی درجی و انتخاب عنصر میانه
- ۰ مرحله ۳: با تابع SELECT بصورت بازگشتی میانهی میانهها را از $\lceil n/5 \rceil$ میانه انتخاب میکنیم
 - (مرحله x: Partition را با استفاده از میانهی میانهها x انجام بده x برابر x امین عنصر x
 - در سمت بالا ہے: اگر i=k پس x جواب است. اگر i=k در سمت پایین و i=k در سمت بالا دنبال عدد بگرد

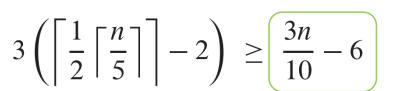
دانشگاه صنعتی امیر کبیر (پلی تکنیک تیوان)

مسئله انتخاب با بدترین زمان اجرای خطی





- دایره: اعداد داخل آرایه
- ستون: گروههای ۵ عنصری
- دایره سفید: میانه هر گروه
 - x: میانهی میانهها x
- پیکان: از بزرگتر به کوچکتر
- x ناحیه خاکستری: اعداد بزرگتر از \star
- حداقل نیمی از میانهها بزرگتر یا مساوی میانهی میانهها x هستند \cdot
 - حداقل نیمی از $\lceil n/5
 ceil$ گروه حداقل ۳ عنصر بزرگتر از x دارند
 - (حداقل) گروه کوچک و گروه خود x را استثنا میکنیم \cdot
 - بالعكس آن هم صادق است! حداقل همين تعداد كوچكتر داريم



مسئله انتخاب با بدترین زمان اجرای خطی



O(n)

مرحله ۱: تقسیم ورودی به $\lfloor n/5
floor$ گروه متشکل از ۵ عنصر و حداکثر یک گروه مابقی اعداد \cdot

O(n)

 \cdot مرحله ۲: یافتن میانه در هر [n/5] گروه - مرتبسازی درجی و انتخاب عنصر میانه

 $T(\lceil n/5 \rceil)$

مرحله m: با تابع SELECT بصورت بازگشتی میانهی میانهها را از $\lfloor n/5 \rfloor$ میانه انتخاب میکنیم

O(n)

(ا با استفاده از میانهی میانهها x انجام بده Partition و مرحله x امین عنصر x

T(7n/10+6) ورحله ه: اگر i=k پس X جواب است. اگر i=k در سمت پایین و i=k در سمت بالا دنبال عدد بگرد i=k

$$T(n) = T(\lceil n/5 \rceil) + T(7n/10 + 6) + O(n)$$

مسئله انتخاب با بدترین زمان اجرای خطی



$$T(n) = T(\lceil n/5 \rceil) + T(7n/10 + 6) + O(n)$$

$$T(n) \le c \lceil n/5 \rceil + c(7n/10 + 6) + an$$

 $\le cn/5 + c + 7cn/10 + 6c + an$
 $= 9cn/10 + 7c + an$
 $= cn + (-cn/10 + 7c + an)$,

$$n \geq 140$$

$$-cn/10 + 7c + an \le 0$$
 $c \ge 10a(n/(n-70))$

$$n/(n-70) \leq 2$$

$$c \geq 20a$$

$$T(n) \le \begin{cases} O(1) & \text{if } n < 140 \\ T(\lceil n/5 \rceil) + T(7n/10 + 6) + O(n) & \text{if } n \ge 140 \end{cases}$$