Sıra İstatistikleri, Ortanca

Sıra İstatistikleri

- Rastgele böl ve fethet
- Beklenen sürenin çözümlemesi
- En kötü durum doğrusal-süre sıra istatistikleri
- Çözümleme

Sıra İstatistikleri

- Doğrusal zaman çözümüne gereksinim duyulur.
- **n** elamanlı bir dizide **i** ‘ inci sıra istatistiği, i’ inci en küçük elemanı bulmak
- i=1 ise *minimum*
- i=n ise *maximum*
- i=n/2 orta değeri (medyan)
  - Eğer *n* tek ise, 2 medyan vardır.
- *Sıra istatistiğini nasıl hesaplayabiliriz?*
- *Çalışma zamanı nedir?*

Sıra İstatistikleri (doğrusal zamanda)

n elemanın i'ninci küçük değerini seçin (i ***ranklı*** eleman).

- $i = 1$: ***minimum***; (en az)
- $i = n$: ***maximum***; (en çok)
- $i = \lfloor (n+1)/2 \rfloor$ veya $\lceil (n+1)/2 \rceil$: ***median***. (ortanca)

Saf algoritma: i'ninci elemanı sırala ve dizinle.

En kötü koşma süresi $= \Theta(n \lg n) + \Theta(1)$
$$= \Theta(n \lg n),$$
birleştirme veya yığin sıralaması kullan (*çabuk sıralamayı değil*).

Sıra istatistiklerinin Bulunması: Seçim Problemi

- **Seçme daha ilginç problemdir:** Bir dizideki i' inci en küçük elemanı bulma. Bunun için iki algoritma;
  - Beklenen çalışma zamanı O(n) olan bir pratik rastgele algoritması
  - En kötü çalışma zamanı sadece O(n) ile ilgili teorik algoritma
- **Anahtar Fikir:** Quicksort algoritmasındaki rastgele bölüntüyü kullanmak.
  - Fakat, sadece bir altdizi incelememiz gerekir
  - **Bu işlem çalışma zamanında tasarruf sağlar: O(n)**

Rastgele böl-ve-fethet algoritması

RAND-SELECT(A, p, q, i) $\triangleleft A[p\,.\,q]$'nın i'ninci en küçüğü

  if $p = q$ **then return** $A[p]$

  $r \leftarrow$ RAND-PARTITION(A, p, q) (Rastgele bölüntü)

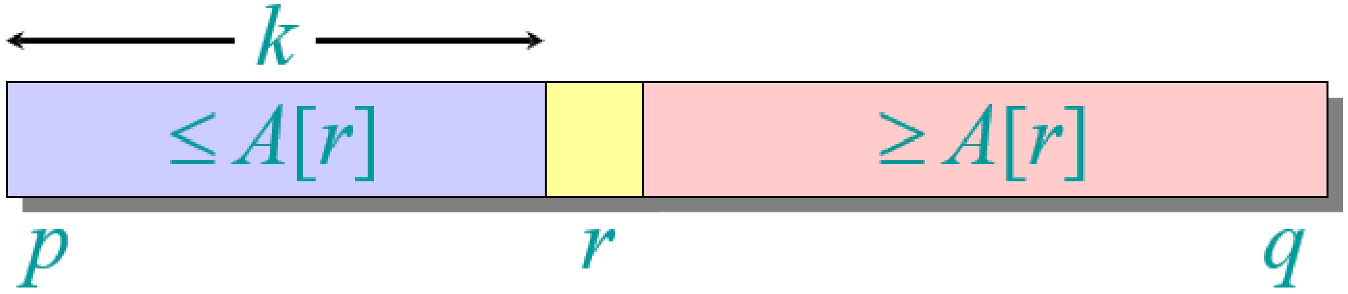
  $k \leftarrow r - p + 1$ $\triangleleft k = \text{rank}(A[r])$ (rütbeli)

  if $i = k$ **then return** $A[r]$

  if $i < k$

    then return RAND-SELECT$(A, p, r-1, i)$

    else return RAND-SELECT$(A, r+1, q, i-k)$



Örnek

$i = 7$' nci en küçük olarak seçin:

6	10	13	5	8	3	2	11	$i = 7$

pivot ***(esas eleman)***

Partition (Bölüntü):

2	5	3	6	8	13	10	11	$k = 4$

$7 - 4 = 3$ 'üncü küçüğü özyinelemeyle seçin.

Rastgele Seçme Analizi: Çözümlemede sezgi (öngörü)

- (Çözümlemelerin hepsinde tüm elemanların farklı olduğu varsayılıyor.)

Şanslı durum:

$$T(n) = T(9n/10) + \Theta(n)$$
$$= \Theta(n)$$

En iyi durum (Best case):
9:1 bölüntü (partion)
olduğunu farz edin

$$n^{\log_{10/9} 1} = n^0 = 1$$

DURUM 3

Şanssız durum:

$$T(n) = T(n-1) + \Theta(n)$$
$$= \Theta(n^2)$$

aritmetik seri

Sıralamadan daha kötü!

En kötü durum(Worst case:
Bölüntü daima 0:n-1

Beklenen süre çözümlemesi

Çözümleme rastgele çabuk sıralamanın benzeri ama bazı farkları var.

$T(n)$ = Rastgele-seçim koşma süresinin rastgele değişkeni olsun (n boyutlu bir girişte), ve rastgele sayılar birbirinden bağımsız olsun.

$k = 0, 1, \ldots, n–1$ için ***göstergesel rastgele değişkeni*** tanımlayın.

$$X_k = \begin{cases} 1 & \text{eğer BÖLÜNTÜ } k : n–k–1 \text{ bölmeli ise,} \\ 0 & \text{diğer durumlarda.} \end{cases}$$

Çözümleme (devam)

Bir üst sınır elde etmek için, i' ninci elemanın her zaman bölüntünün büyük bölgesinde olduğunu varsayın:

$$T(n) = \begin{cases} T(\max\{0, n-1\}) + \Theta(n), & 0 : n-1 \text{ bölünmesi}, \\ T(\max\{1, n-2\}) + \Theta(n), & 1 : n-2 \text{ bölünmesi}, \\ \quad\vdots & \\ T(\max\{n-1, 0\}) + \Theta(n), & n-1 : 0 \text{ bölünmesi}, \end{cases}$$

$$= \sum_{k=0}^{n-1} X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right).$$

Beklenenin hesaplanması

$$E[T(n)] = E\left[\sum_{k=0}^{n-1} X_k\left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E\left[X_k\left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

Beklenenin doğrusallığı.

Beklenenin hesaplanması

$$E[T(n)] = E\left[\sum_{k=0}^{n-1} X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E\left[X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E[X_k] \cdot E[T(\max\{k, n-k-1\}) + \Theta(n)]$$

$$= \frac{1}{n} \sum_{k=0}^{n-1} E[T(\max\{k, n-k-1\})] + \frac{1}{n} \sum_{k=0}^{n-1} \Theta(n)$$

Beklenenin doğrusallığı; $E[X_k] = 1/n$.

Beklenenin hesaplanması

$$E[T(n)] = E\left[\sum_{k=0}^{n-1} X_k\left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E\left[X_k\left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E[X_k] \cdot E[T(\max\{k, n-k-1\}) + \Theta(n)]$$

$$= \frac{1}{n}\sum_{k=0}^{n-1} E[T(\max\{k, n-k-1\})] + \frac{1}{n}\sum_{k=0}^{n-1}\Theta(n)$$

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

k=0, için T(max{0,n-0-1}
k=n-1 için T(max{n-1,0}
Dolayısıyla yarıya kadar toplayıp 2 ile çarpınca aynı sonucu elde ederim.

Üstteki terimler
iki kez görünüyor.

Karmaşık yineleme

(Ama çabuk sıralamanınki kadar karmaşık değil.)

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

* Yerine koyma metodu ile kanıtlanır.

Kanıtla: $E[T(n)] \leq c\,n$ sabiti için $c > 0$.

- c sabiti öyle büyük seçilebilir ki, $E[T(n)] \leq c\,n$ tüm taban durumlarında geçerli olur.

Veri: $\sum_{k=\lfloor n/2 \rfloor}^{n-1} k \leq \frac{3}{8} n^2$ (alıştırma).

Yerine koyma metodu

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

Tümevarım hipotezini yerleştirin.

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

$$\le \frac{2c}{n}\left(\frac{3}{8}n^2\right) + \Theta(n)$$

Veriyi kullanın.

Yerine koyma metodu

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

$$\le \frac{2c}{n}\left(\frac{3}{8}n^2\right) + \Theta(n)$$

$$= cn - \left(\frac{cn}{4} - \Theta(n)\right)$$

istenen* – *kalan şeklinde gösterin.

Yerine koyma metodu

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

$$\le \frac{2c}{n}\left(\frac{3}{8}n^2\right) + \Theta(n)$$

$$= cn - \left(\frac{cn}{4} - \Theta(n)\right)$$

$$\le cn\,,$$

c yeterince büyük seçilirse
$cn/4$, $\Theta(n)$' nin üstünde olur.

Rastgele sıra istatistik seçiminin özeti

- Hızlı çalışır: doğrusal beklenen süre.
- Pratikte mükemmel bir algoritma.
- Ama, en kötü durumu ***çok*** kötü: $\Theta(n^2)$.

S. En kötü durumda doğrusal zamanda çalışan bir algoritma var mıdır?

C. Evet, Blum, Floyd, Pratt, Rivest ve Tarjan [1973] sayesinde vardır.

FİKİR: İyi bir pivotu yinelemeyle üretmek.

n=100, n/5= 20, 20/2 = 10 ortanca değer

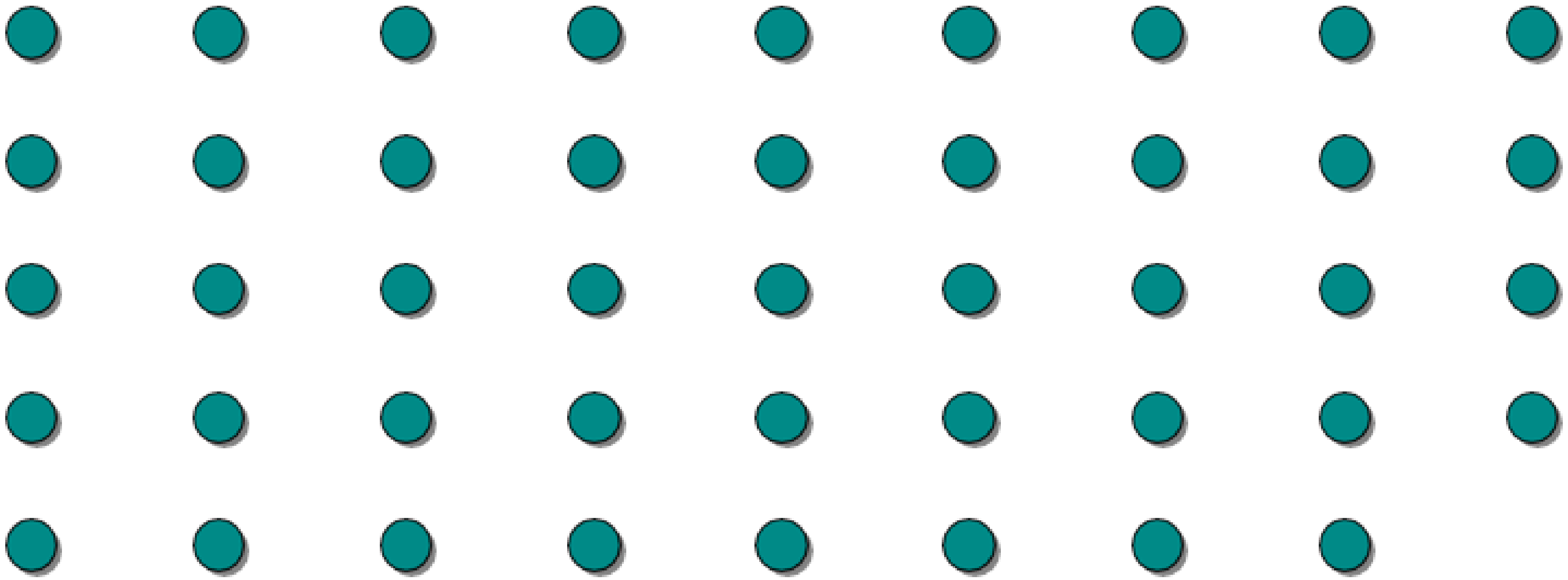
En kötü durum doğrusal-zaman sıra istatistikler

SEÇ (i, n)

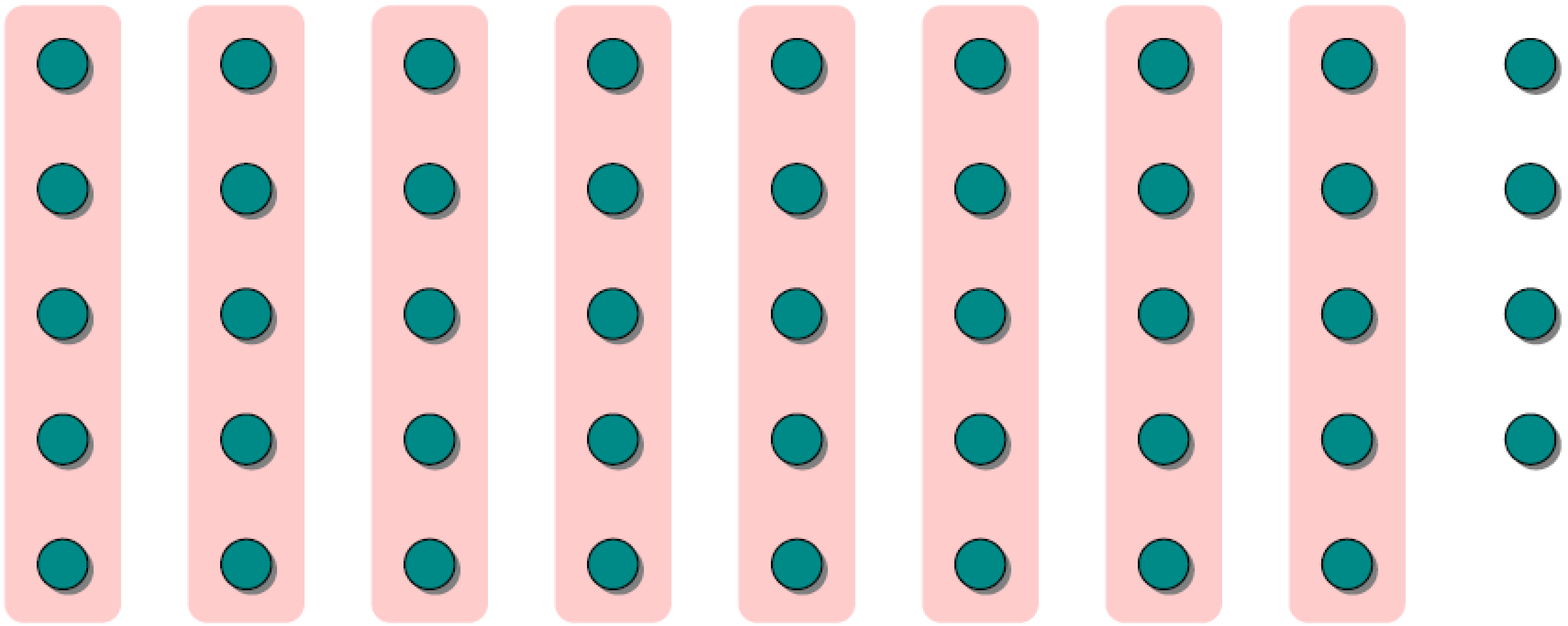
1. n elemanı 5' li gruplara bölün. Her 5' li grubun ortancasını ezbere bulun.
2. $\lfloor n/5 \rfloor$ gruplarının ortancası olacak x' i yinelemeli SEÇME ile pivot olarak belirleyin.
3. Pivot x etrafında bölüntü yapın. $k = \text{rank}(x)$.
   if $i = k$ **then return** x (eğer / öyleyse çıkar)
   elseif $i < k$ (diğer durumlarda)
   then i' ninci en küçük elemanı alt bölgede yinelemeyle SEÇİN.
   else $(i–k)$' nıncı en küçük elemanı üst bölgede yinelemeyle SEÇİN.

RASTGELE-SEÇİMİN aynısı

Pivot seçimi



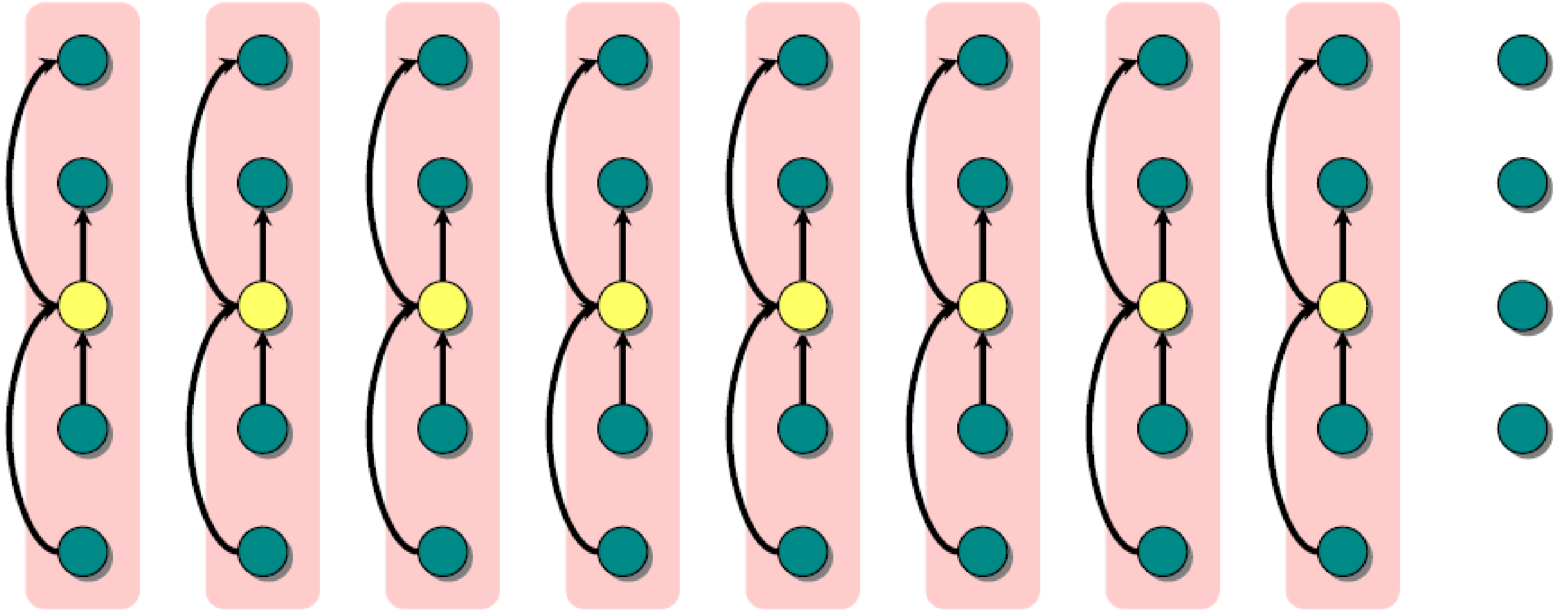
Pivot seçimi



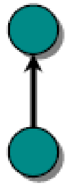
1. n elemanı 5' li gruplara bölün.

* n her zaman tam bölünmeyebilir. Son grupta elemanlar eksik kalabilir bu durumda o sütun dikkate alınmaz.

Pivot seçimi

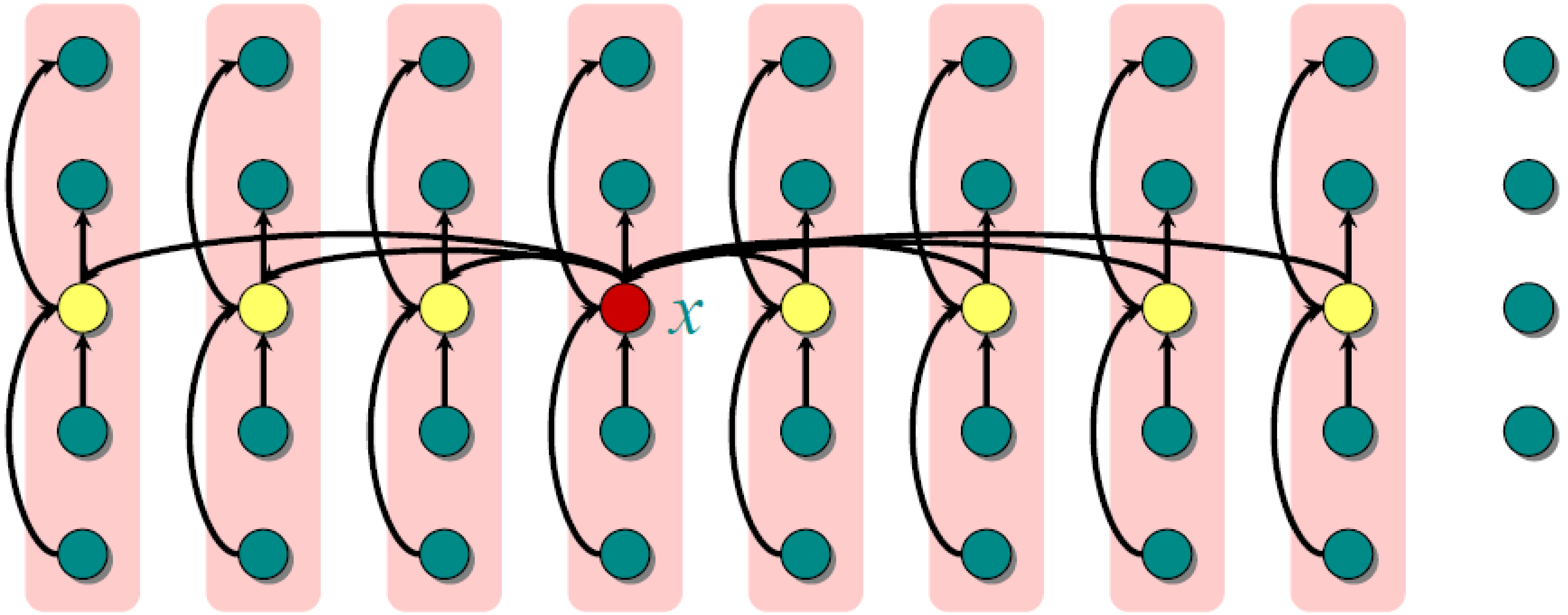


1. n elemanlarını 5' li gruplara bölün. 5-elemanlı grupların ortancasını ezbere bulun.

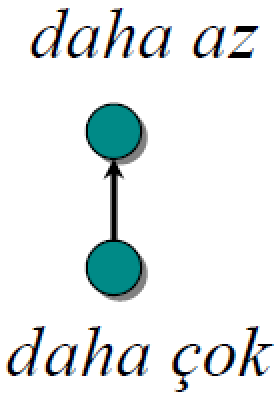


n/5 öbek, her birinde beş eleman var; her birinin ortancasını hesaplamak ne kadar zaman alır?
2 kere n/5. Yani, karşılaştırmaları sayıyorsunuz ve bu Θ (n) ‘dir.
Sonuçta her grupta 5 sayı var ve sabit sayıda karşılaştırma yapılır.

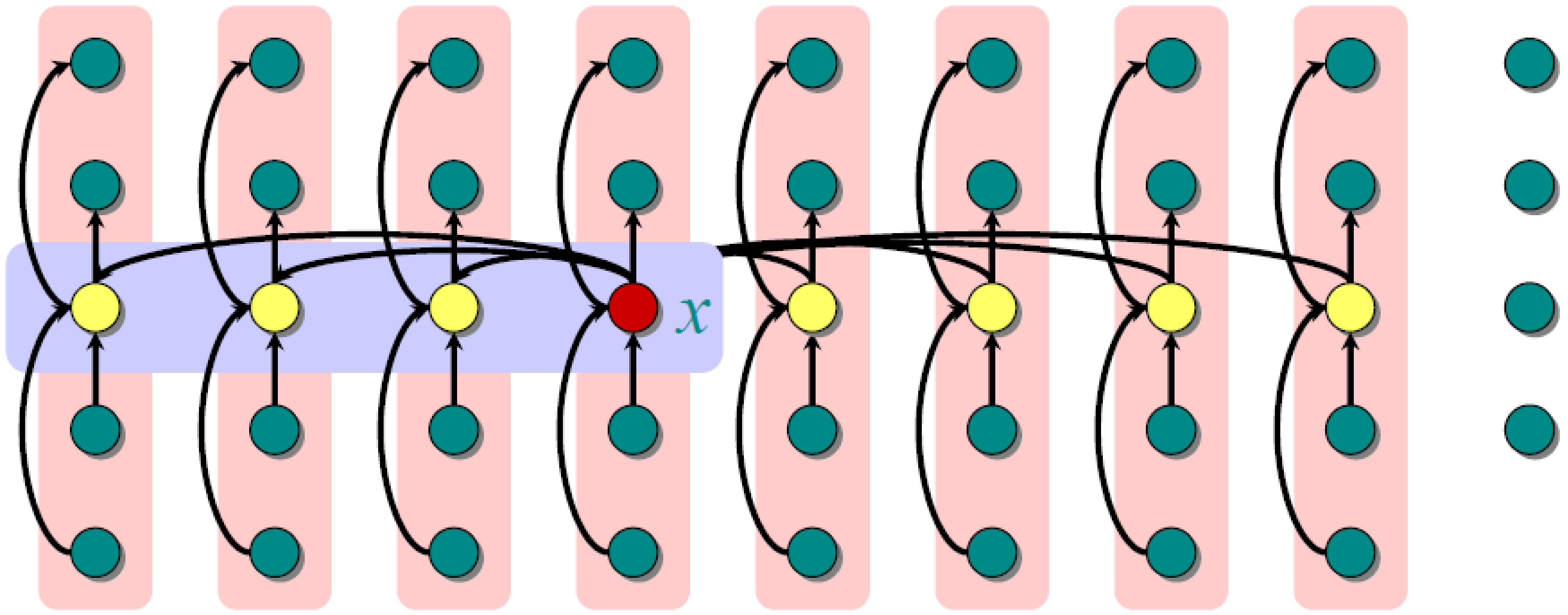
Pivot seçimi



1. n elemanlarını 5' li gruplara bölün. 5 elemanlı grupların ortancalarını ezbere bulun.
2. $\lfloor n/5 \rfloor$ gruplarının ortancası olacak x' i, yinelemeli SEÇME ile pivot olarak belirleyin.

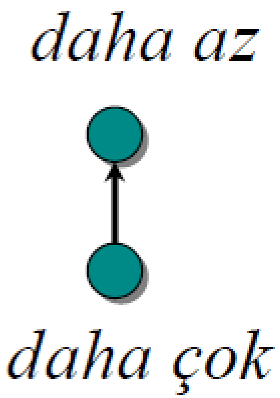


Çözümleme

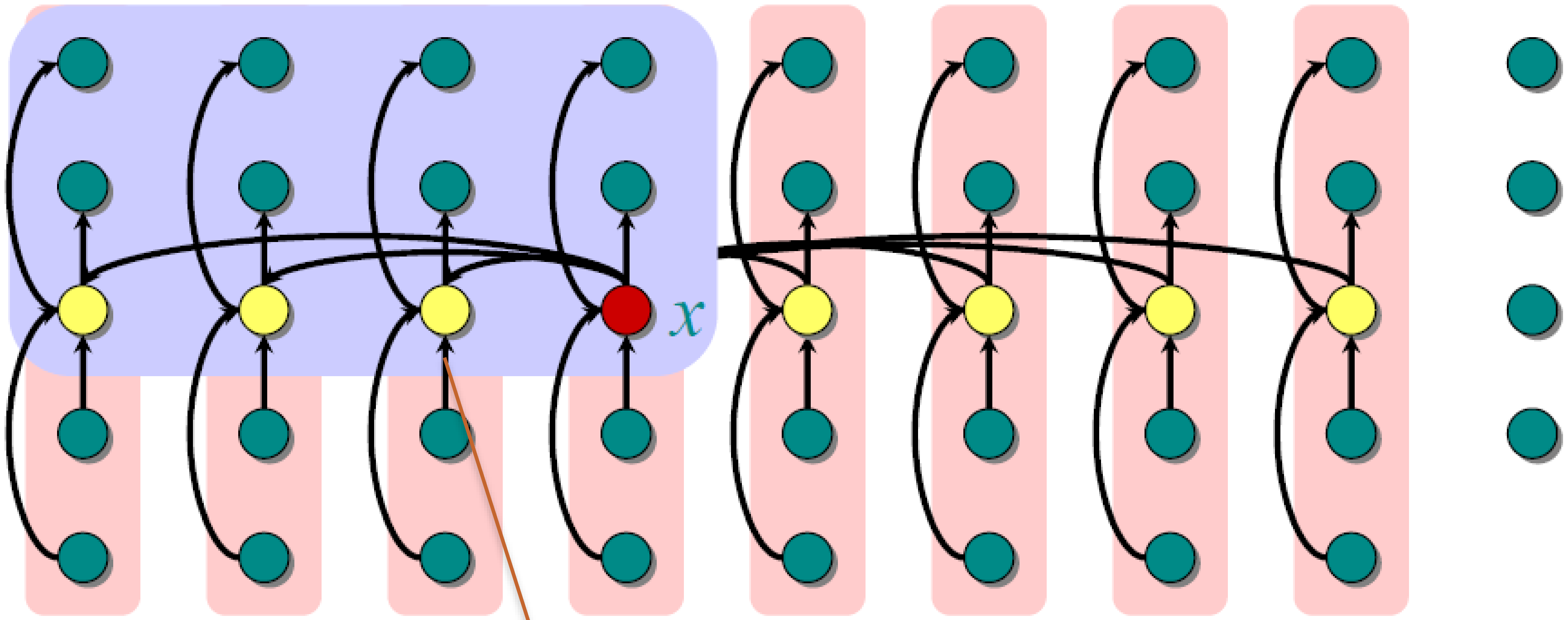


Grup ortancalarının en az yarısı $\leq x$, bu da en az $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ grup ortancası eder.

n=100, n/5= 20, 20/2 = 10 ortanca değer

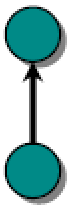


Çözümleme (Tüm elemanları farklı varsay.)

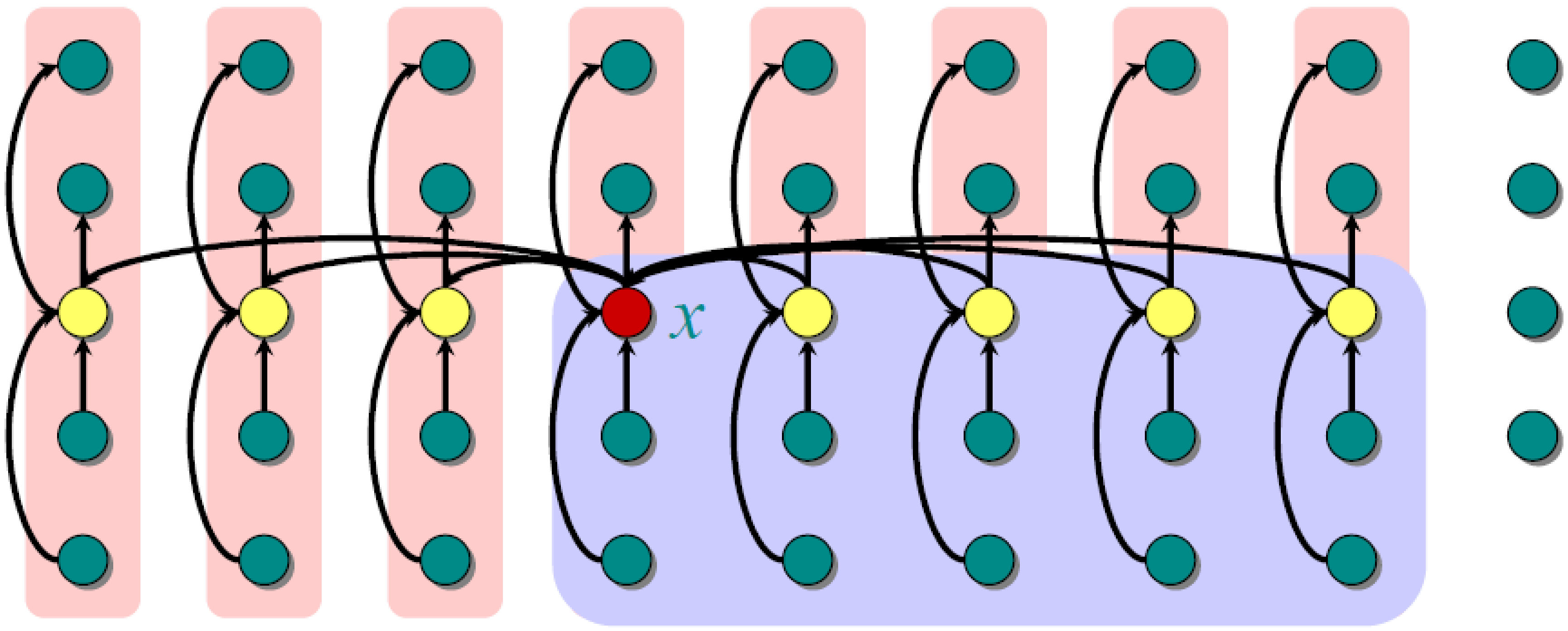


Grup ortancalarının en az yarısı $\leq x$, bu da en az $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ grup ortancası eder.

- Bu nedenle, en az $3\lfloor n/10 \rfloor$ eleman $\leq x$.

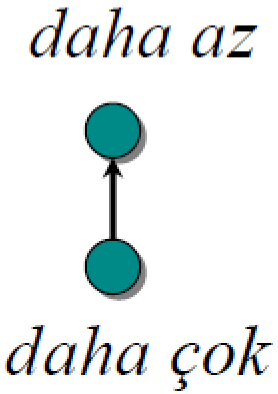


Çözümleme (Tüm elemanları farklı varsay.)



Grup ortancalarının en az yarısı $\leq x$, bu da en az $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ grup ortancası eder.

- Bu nedenle, en az $3 \lfloor n/10 \rfloor$ eleman $\leq x$.
- Benzer şekilde, en az $3 \lfloor n/10 \rfloor$ eleman $\geq x$.



Önemsiz basitleştirme

- $n \geq 50$ için, $3\lfloor n/10 \rfloor \geq n/4$ olur.
- Bu nedenle, $n \geq 50$ için Adım 4'teki SEÇİM özyinelemeli olarak $\leq 3n/4$ eleman kapsamında yapılır.
- Böylece, koşma süresinin yinelemesinde Adım 4'ün en kötü durumda $T(3n/4)$ zamanı alacağı farz edilebilir.
- $n < 50$ için en kötü sürenin $T(n) = \Theta(1)$ olduğunu biliyoruz.

5'li grupta 4 çeyrek olduğundan en az n/4 lük kısımda en fazla ise 3n/4'lük kısım ile uğraşılır.

Yinelemeyi geliştirmek

$T(n)$ SELECT (i, n) (SEÇİN)

$\Theta(n)$ 1. n elemanı 5' li gruplara ayırın. 5-elemanlı grupların ortancasını ezberden bulun.

$(n/5)$ 2. $\lfloor n/5 \rfloor$ gruplarının ortancası olacak x' i, yinelemeli SEÇME ile pivot olarak belirleyin.

$\Theta(n)$ 3. Pivot x etrafında bölüntü yapın. $k = \text{rank}(x)$ olsun.

$T(3n/4)$ 4. **if** $i = k$ **then return** x
elseif $i < k$
then i' ninci en küçük elemanı alt bölgede yinelemeli olarak SEÇİN.
else $(i–k)$'nıncı en küçük elemanı üst bölgede yinelemeli olarak SEÇİN.

Yinelemeyi çözmek

$$T(n) = T\left(\frac{1}{5}n\right) + T\left(\frac{3}{4}n\right) + \Theta(n)$$

Yerine koyma:

$T(n) \leq cn$

$$\begin{aligned} T(n) &\leq \frac{1}{5}cn + \frac{3}{4}cn + \Theta(n) \\ &= \frac{19}{20}cn + \Theta(n) \\ &= cn - \left(\frac{1}{20}cn - \Theta(n)\right) \\ &\leq cn \quad , \end{aligned}$$

c, hem $\Theta(n)$' i hem de başlangıç koşullarını gözeterek yeterince büyük seçilirse...

Sonuçlar

- Yinelemenin her düzeyindeki iş sabit bir kesir (19/20) oranında küçüldüğünden, düzeylerdeki iş bir geometrik seri gibidir ve kökteki doğrusal iş ön plana çıkar.
- Pratikte bu algoritma yavaş çalışır, çünkü n' nin önündeki sabit büyüktür.
- Rastgele algoritma çok daha pratiktir.

Alıştırma: *Neden 3' lü gruplara bölmüyoruz?*

Kaynakça

- Algoritmalar : Prof. Dr. Vasif NABİYEV, Seçkin Yayıncılık
- Algoritmalara Giriş : Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Palme YAYINCILIK
- Algoritmalar : Robert Sedgewick , Kevin Wayne, Nobel Akademik Yayıncılık
- M.Ali Akcayol, Gazi Üniversitesi, Algoritma Analizi Ders Notları
- Doç. Dr. Erkan TANYILDIZI, Fırat Üniversitesi, Algoritma Analizi Ders Notları
- http://www.bilgisayarkavramlari.com