بسم الله الرحمن الرحیم

# جلسه هشتم

## خلاصه‌سازی برای مه‌داده

# خلاصه‌سازی گراف

# زیرمسئله ۱: بازیابی K_ تنک

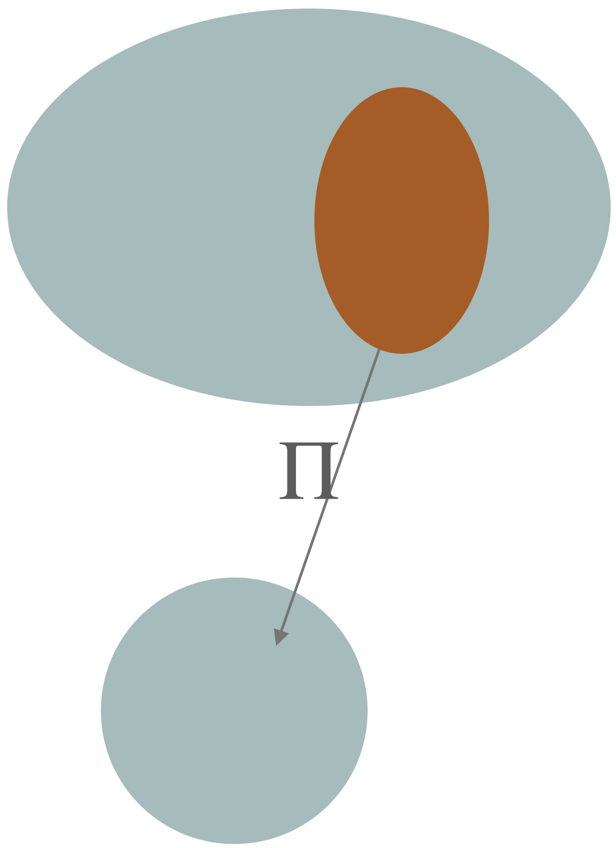
- $x$: $k$_ تنک
  - چرا روش معمول کار نمی‌کند؟

# زیرمسئله ۱: بازیابی K_ تنک

- x: k_ تنک
  - چرا روش معمول کار نمی‌کند؟
- نگه‌داری $\Pi x$ به جای x

# زیرمسئله ۱: بازیابی K_تنک

- x: k_تنک
  - چرا روش معمول کار نمی‌کند؟




- نگه‌داری $\Pi x$ به جای x
- قابل بازیابی؟
  - به ازای هر k_تنک x و y: $\Pi x \neq \Pi y$
  - به ازای هر 2k_تنک z: $\Pi z \neq 0$

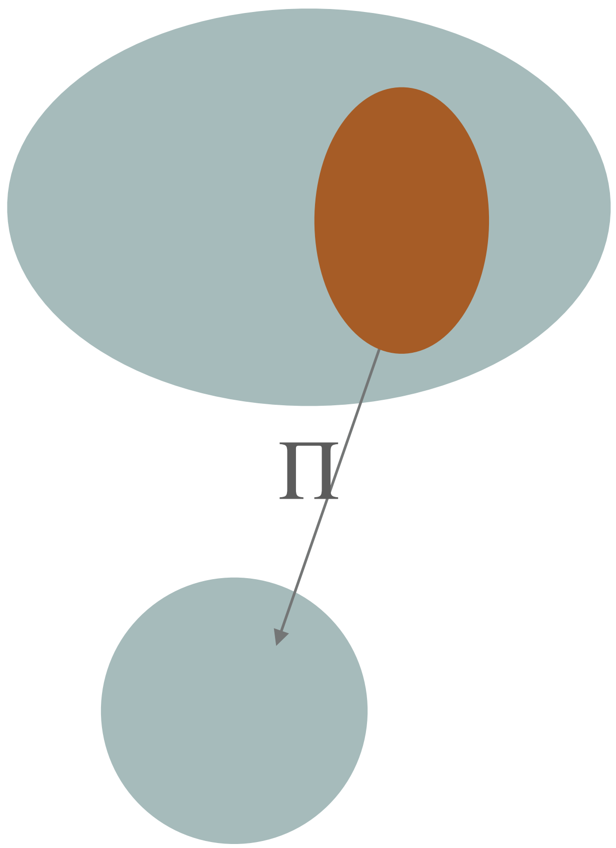
# زیرمسئله ۱: بازیابی K_تنک

- $x$: $k$_ تنک

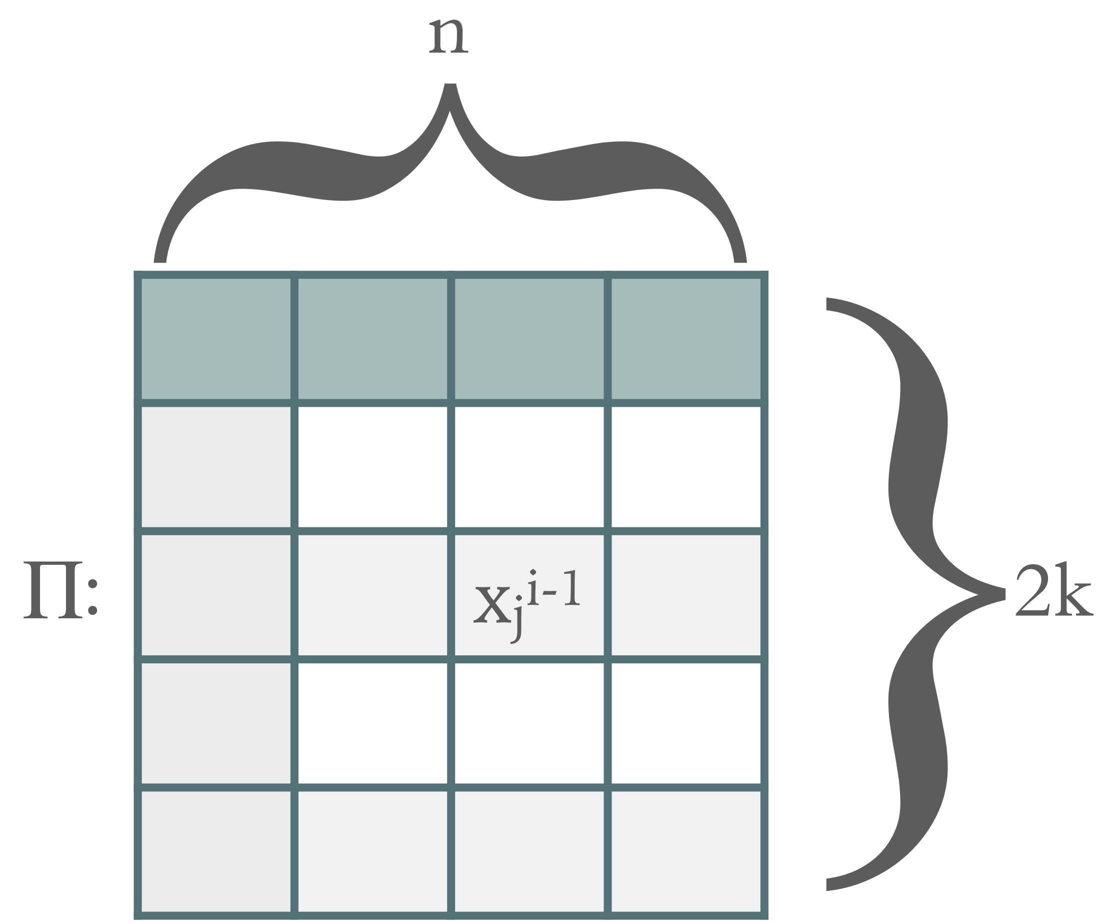
  - چرا روش معمول کار نمی‌کند؟

# زیرمسئله ۱: بازیابی K_تنک

- x: k_تنک
  - چرا روش معمول کار نمی‌کند؟




- نگه‌داری $\Pi x$ به جای x
- قابل بازیابی؟
  - به ازای هر k_تنک x و y: $\Pi x \neq \Pi y$
  - به ازای هر 2k_تنک z: $\Pi z \neq 0$




**Fact 4.2.1.** *Let $A \in F^{r \times r}$ be such that $A_{i,j} = x_j^{i-1}$ for $i, j \in [r]$ for some field $F$. Then*

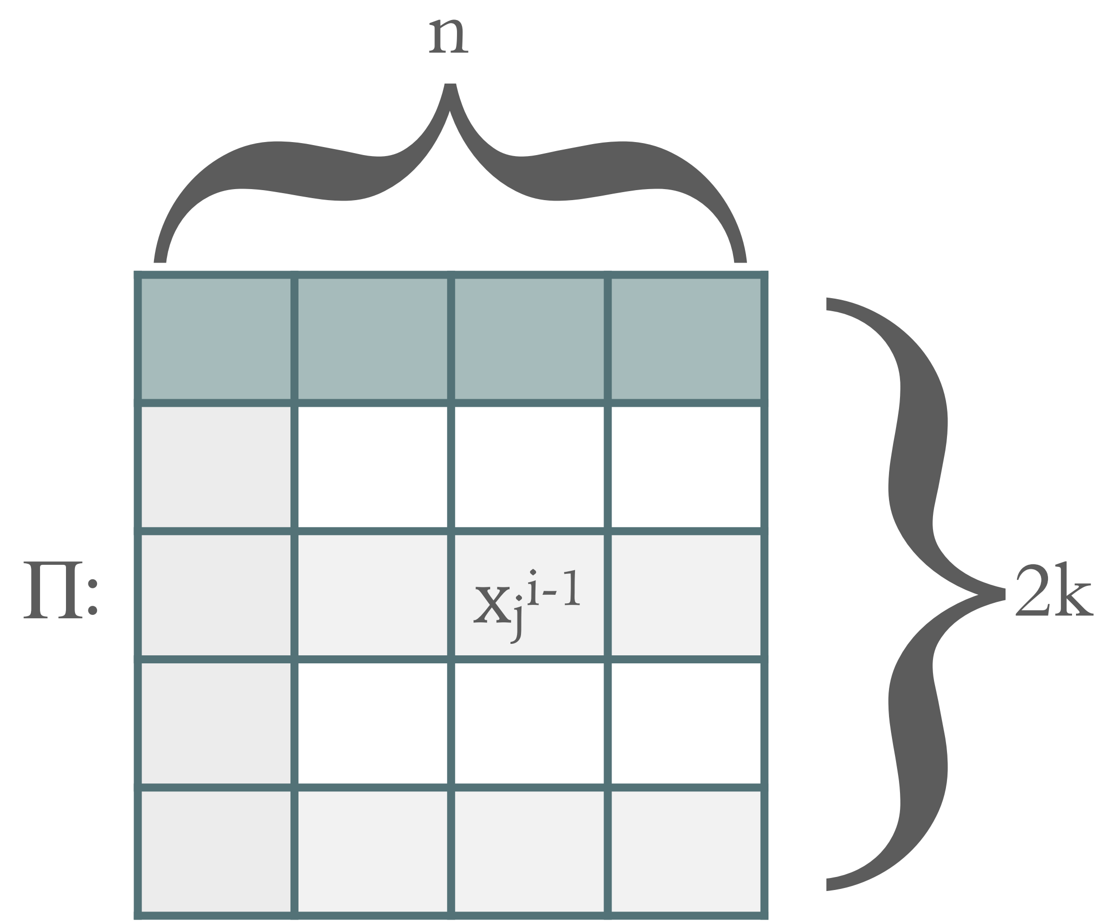
$$\det(A) = \prod_{1 \le i < j \le n} (x_i - x_j)$$

## زیرمسئله ۱: بازیابی K_تنک

حافظه: 2k log p

احتمال ۱

زمان: $O(k^2 \, polylog(p))$




**Fact 4.2.1.** *Let $A \in F^{r \times r}$ be such that $A_{i,j} = x_j^{i-1}$ for $i, j \in [r]$ for some field $F$. Then*

$$\det(A) = \prod_{1 \leq i < j \leq n} (x_i - x_j)$$

# زیرمسئله ۱: بازیابی Kـتنک

حافظه: 2k log p

احتمال ۱

زمان: $O(k^2\, polylog(p))$

# زیرمسئله۲: SUPPORT-FIND

| | |
|---|---|
| 1 | $A_1$ |
| 2 | $A_2$ |
| … | … |
| log n | $A_{\log n}$ |

- تعریف مسئله: با احتمال $\delta$ یک $i \in \text{support}(x)$
- الگوریتم:

ایده

- شروع:
- h: تابع درهم‌ساز نمایی $\mathbb{P}(h(i) = j) = 1/2^j$
- $A_j$ (بازیابی k_تنک) برای $j \in [\log_2 n]$
- $k = C \log(1/\delta)$

# زیرمسئله۲: SUPPORT-FIND

| | |
|---|---|
| 1 | $A_1$ |
| 2 | $A_2$ |
| … | … |
| log n | $A_{\log n}$ |

- تعریف مسئله: با احتمال $\delta$ یک $i \in \text{support}(x)$
- الگوریتم:

ایده

- شروع:
  - h: تابع درهم‌ساز نمایی $\mathbb{P}(h(i) = j) = 1/2^j$
  - $A_j$ (بازیابی k_تنک) برای $j \in [\log_2 n]$
    - $k = C \log(1/\delta)$
- update(i,Δ): $A_{h(i)}.\mathsf{update}(i, \Delta)$

# زیرمسئله۲: SUPPORT-FIND

| | |
|---|---|
| 1 | $A_1$ |
| 2 | $A_2$ |
| ... | ... |
| log n | $A_{\log n}$ |

- تعریف مسئله: با احتمال $\delta$ یک $i \in \text{support}(x)$
- الگوریتم:

ایده

  - شروع:
    - $h$: تابع درهم‌ساز نمایی $\mathbb{P}(h(i) = j) = 1/2^j$
    - $A_j$ (بازیابی k_تنک) برای $j \in [\log_2 n]$
      - $k = C \log(1/\delta)$
  - update(i,Δ): $A_{h(i)}.\mathsf{update}(i, \Delta)$
  - جواب:
    - اولین بزرگ‌ترین j که $A_j$ خالی نبود: یکی از اعضای $A_j$

# زیرمسئله۲: SUPPORT-FIND

| | |
|---|---|
| 1 | $A_1$ |
| 2 | $A_2$ |
| … | … |
| log n | $A_{\log n}$ |

- تعریف مسئله: با احتمال $\delta$ یک $i \in \text{support}(x)$
- الگوریتم:

ایده

  - شروع:
- h: تابع درهم‌ساز نمایی $\mathbb{P}(h(i) = j) = 1/2^j$
- $A_j$ (بازیابی k_تنک) برای $j \in [\log_2 n]$
- $k = C \log(1/\delta)$

# تحلیل SUPPORT-FIND:

- $x=0$: درست
- $|\text{support}(x)| \leq k$ : درست

# تحلیل SUPPORT-FIND:

- $x=0$: درست
- $|\,support(x)\,| \le k$ : درست
- پس: $k < t =: |\,support(x)\,| \quad (k := C \log(1/\delta))$

# تحلیل SUPPORT-FIND:

- $x=0$: درست
- $|\,support(x)\,| \le k$ : درست
- پس: $k < t =: |\,support(x)\,|$ $\quad (k := C \log(1/\delta))$
- $T_j$: تعداد $x_i$های غیر صفر که در $A_j$ می‌افتند
- $ET_j = t/2^j$

# تحلیل SUPPORT-FIND:

- $x=0$: درست
- درست : $|\,\mathrm{support}(x)\,| \le k$
- پس: $|\,\mathrm{support}(x)\,| := t > k \quad (k := C\log(1/\delta))$
- $T_j$: تعداد $x_i$های غیر صفر که در $A_j$ می‌افتند
  - $ET_j = t/2^j$
- $j^*$: $c\log(1/\delta) \le t/2^{j^*} < 2c\log(1/\delta)$ $\quad 1 < c < C$

# تحلیل SUPPORT-FIND:

- $x=0$: درست
- $|\text{support}(x)| \leq k$ : درست
- پس: $|\text{support}(x)| := t > k \quad (k := C\log(1/\delta))$
- $T_j$: تعداد $x_i$های غیر صفر که در $A_j$ می‌افتند
  - $ET_j = t/2^j$
- $j^*$: $c\log(1/\delta) \leq t/2^{j^*} < 2c\log(1/\delta)$ $\quad 1 < c < C$
- اتفاق‌های تضمین کننده
  - $\mathcal{E}_1$: $\max_{j \geq j^*} T_j \leq k$
  - $\mathcal{E}_2$: $T_{j^*} \geq 1$

چرا؟

# تحليل SUPPORT-FIND:

- $\mathcal{E}_1$: $\max_{j \geq j^*} T_j \leq k$

$$\mathbb{P}(\neg\mathcal{E}_1) \leq \sum_{j=j^*}^{\log n} \mathbb{P}(T_j > k)$$

# تحليل SUPPORT-FIND:

- $\mathcal{E}_1$ : $\max_{j \geq j^*} T_j \leq k$

$$\mathbb{P}(\neg\mathcal{E}_1) \leq \sum_{j=j^*}^{\log n} \mathbb{P}(T_j > k)$$

$$= \mathbb{P}(T_j > (k2^j/t) \cdot \mathbb{E}\, T_j)$$

$ET_j = t/2^j$

# تحلیل SUPPORT-FIND:

- $\mathcal{E}_1$ : $\max_{j \geq j^*} T_j \leq k$

$$\mathbb{P}(\neg\mathcal{E}_1) \leq \sum_{j=j^*}^{\log n} \mathbb{P}(T_j > k)$$

$$= \mathbb{P}(T_j > (k2^j/t) \cdot \mathbb{E}\, T_j)$$

$$ET_j = t/2^j$$

$$< (k2^j/t)^{-C'k}$$

# تحليل SUPPORT-FIND:

- $\mathcal{E}_1$ : $\max_{j \geq j^*} T_j \leq k$

$$\mathbb{P}(\neg\mathcal{E}_1) \leq \sum_{j=j^*}^{\log n} \mathbb{P}(T_j > k)$$

$$= \mathbb{P}(T_j > (k2^j/t) \cdot \mathbb{E}\, T_j)$$

$$ET_j = t/2^j$$

$$< (k2^j/t)^{-C'k}$$

$$\mathbb{P}(X > (1+\lambda)\mu) < \left(\frac{e^{\lambda}}{(1+\lambda)^{1+\lambda}}\right)^{\mu} < \lambda^{-\Omega(\lambda\mu)} < \sqrt{1/(1+\lambda)}^{\,-\Omega(\lambda\mu)}$$

$(\lambda > 2e - 1)$

# تحليل SUPPORT-FIND:

- $\mathcal{E}_1$ : $\max_{j \geq j^*} T_j \leq k$

$$\mathbb{P}(\neg\mathcal{E}_1) \leq \sum_{j=j^*}^{\log n} \mathbb{P}(T_j > k)$$

$$= \mathbb{P}(T_j > (k2^j/t) \cdot \mathbb{E}\, T_j)$$

$ET_j = t/2^j$

$$< (k2^j/t)^{-C'k}$$

$$\mathbb{P}(X > (1+\lambda)\mu) < \left(\frac{e^\lambda}{(1+\lambda)^{1+\lambda}}\right)^\mu < \lambda^{-\Omega(\lambda\mu)} < \sqrt{1/(1+\lambda)}^{\,-\Omega(\lambda\mu)}$$

$(\lambda > 2e - 1)$

$$c\log(1/\delta) \leq t/2^{j^*} < 2c\log(1/\delta)$$

$k := C\log(1/\delta)$

$C/2c < k2^{j^*}/t \leq C/c$

# تحليل SUPPORT-FIND:

- $\mathcal{E}_1$: $\max_{j \geq j^*} T_j \leq k$

$$\mathbb{P}(\neg\mathcal{E}_1) \leq \sum_{j=j^*}^{\log n} \mathbb{P}(T_j > k) < (C/2c)^{-C'k}$$

$$= \mathbb{P}(T_j > (k2^j/t) \cdot \mathbb{E}\, T_j)$$

$$ET_j = t/2^j$$

$$< (k2^j/t)^{-C'k}$$

$$\mathbb{P}(X > (1+\lambda)\mu) < \left(\frac{e^\lambda}{(1+\lambda)^{1+\lambda}}\right)^\mu < \lambda^{-\Omega(\lambda\mu)} < \sqrt{1/(1+\lambda)}^{-\Omega(\lambda\mu)}$$

$(\lambda > 2e - 1)$

$$c \log(1/\delta) \leq t/2^{j^*} < 2c \log(1/\delta)$$

$k := C \log(1/\delta)$

$C/2c < k2^{j^*}/t \leq C/c$

# تحليل SUPPORT-FIND:

- $\mathcal{E}_1$ : $\max_{j \geq j^*} T_j \leq k$

$$\mathbb{P}(\neg\mathcal{E}_1) \leq \sum_{j=j^*}^{\log n} \mathbb{P}(T_j > k) < (C/2c)^{-C'k} < \delta/2$$

$$= \mathbb{P}(T_j > (k2^j/t) \cdot \mathbb{E}\, T_j)$$

$ET_j = t/2^j$

$$< (k2^j/t)^{-C'k}$$

$$\mathbb{P}(X > (1+\lambda)\mu) < \left(\frac{e^\lambda}{(1+\lambda)^{1+\lambda}}\right)^\mu < \lambda^{-\Omega(\lambda\mu)} < \sqrt{1/(1+\lambda)}^{-\Omega(\lambda\mu)}$$

$(\lambda > 2e - 1)$

$$c \log(1/\delta) \leq t/2^{j^*} < 2c \log(1/\delta)$$

$k := C \log(1/\delta)$

$C/2c < k2^{j^*}/t \leq C/c$

# تحلیل SUPPORT-FIND: (ادامه)

- $\mathcal{E}_2$: $T_{j^*} \geq 1$

$$\mathbb{P}(\neg\mathcal{E}_2) = \mathbb{P}(T_{j^*} = 0)$$

# تحلیل SUPPORT-FIND: (ادامه)

- $\mathcal{E}_2$: $T_{j^*} \geq 1$

$$\mathbb{P}(\neg\mathcal{E}_2) = \mathbb{P}(T_{j^*} = 0) = (1 - \frac{1}{2^{j^*}})^t$$

# تحلیل SUPPORT-FIND: (ادامه)

- $\mathcal{E}_2$: $T_{j*} \geq 1$

$$\mathbb{P}(\neg\mathcal{E}_2) = \mathbb{P}(T_{j*} = 0) = (1 - \frac{1}{2^{j*}})^t$$

$$\leq exp(-t/2^{j*})$$

# تحلیل SUPPORT-FIND: (ادامه)

- $\mathcal{E}_2$: $T_{j^*} \geq 1$

$$\mathbb{P}(\neg\mathcal{E}_2) = \mathbb{P}(T_{j^*} = 0) = (1 - \frac{1}{2^{j^*}})^t$$

$$\leq exp(-t/2^{j^*}) \leq \delta^c$$

$$c\log(1/\delta) \leq t/2^{j^*} < 2c\log(1/\delta)$$

# تحلیل SUPPORT-FIND: (ادامه)

- $\mathcal{E}_2$: $T_{j^*} \geq 1$

$$\mathbb{P}(\neg\mathcal{E}_2) = \mathbb{P}(T_{j^*} = 0) = (1 - \frac{1}{2^{j^*}})^t$$

$$\leq exp(-t/2^{j^*}) \leq \delta^c \qquad \leq \delta/2$$

$$c\log(1/\delta) \leq t/2^{j^*} < 2c\log(1/\delta)$$

$\delta < ½$

# تحلیل SUPPORT-FIND: (ادامه)

- $\mathcal{E}_2$: $T_{j*} \geq 1$

$$\mathbb{P}(\neg\mathcal{E}_2) = \mathbb{P}(T_{j*} = 0) = (1 - \frac{1}{2^{j*}})^t$$

$$\leq exp(-t/2^{j*}) \leq \delta^c \qquad \leq \delta/2$$

$$c\log(1/\delta) \leq t/2^{j*} < 2c\log(1/\delta)$$

$\delta < ½$

- پس: احتمال بد بودن $< \delta$

# تحلیل SUPPORT-FIND: (ادامه)

- $\mathcal{E}_2$: $T_{j^*} \geq 1$

$$\mathbb{P}(\neg\mathcal{E}_2) = \mathbb{P}(T_{j^*} = 0) = \left(1 - \frac{1}{2^{j^*}}\right)^t$$

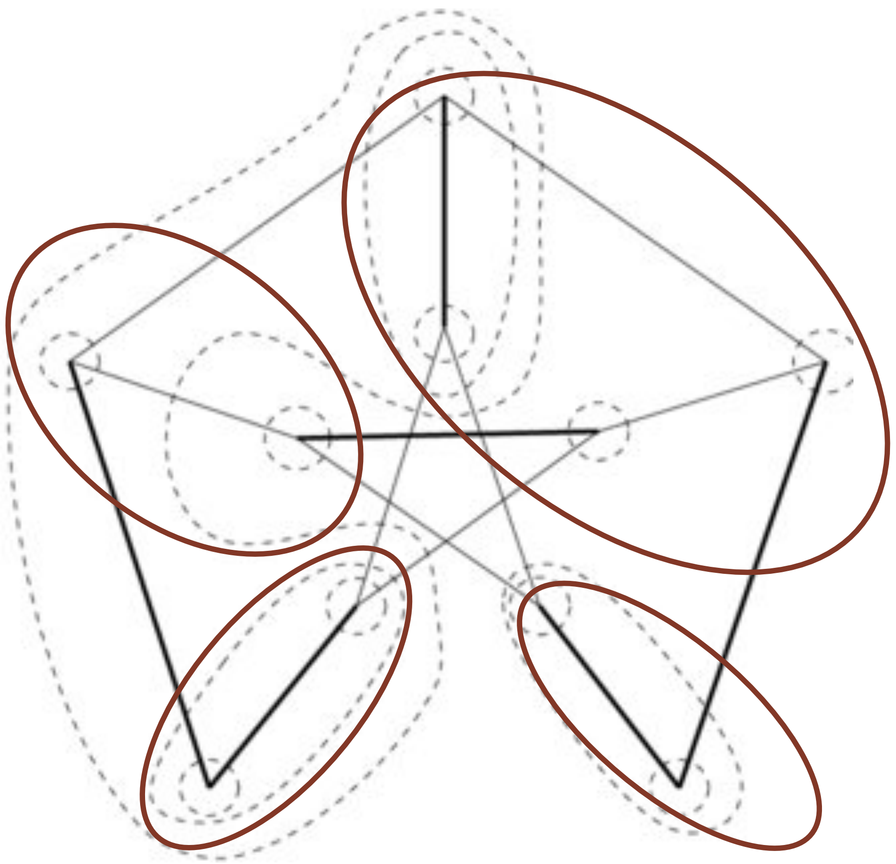
$\leq exp(-t/2^{j^*}) \leq \delta^c$ $\leq \delta/2$

$c\log(1/\delta) \leq t/2^{j^*} < 2c\log(1/\delta)$

$\delta < ½$

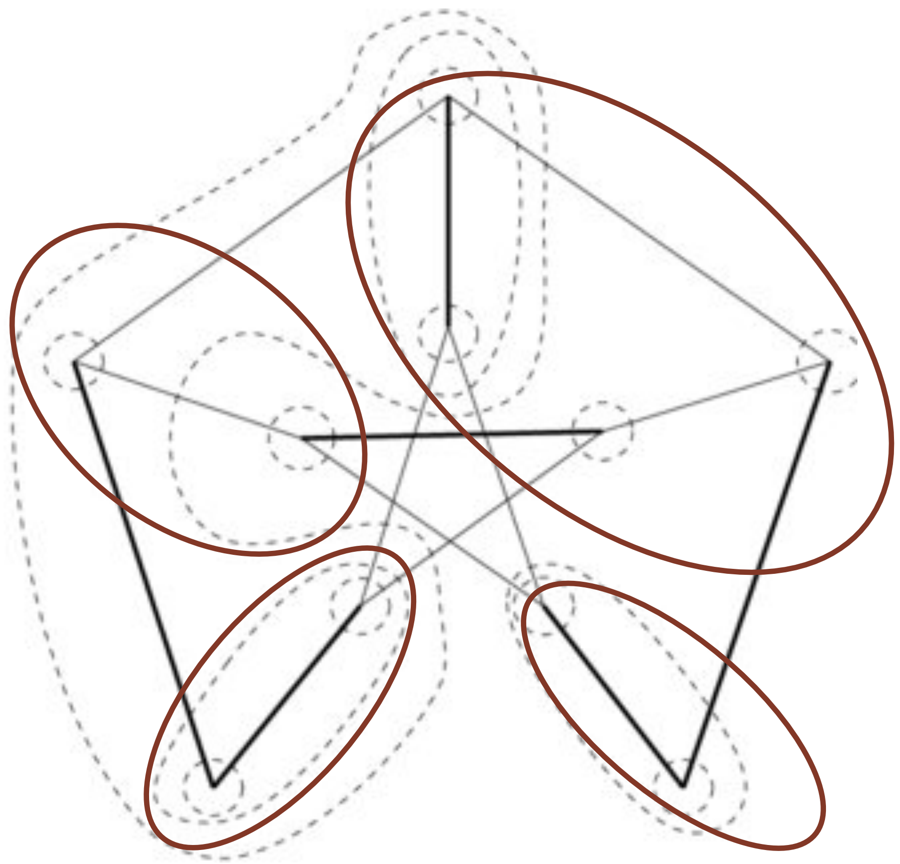
- پس: احتمال بد بودن $< \delta$
- حافظه: $O(\log(1/\delta)\log^2 n)$
  - $\log n$ تا بازیابی k_تنک
  - بازیابی k_تنک: $\log(1/\delta)$ حافظه هر کدام $\log n$ بیت

# مسئله جنگل فراگیر



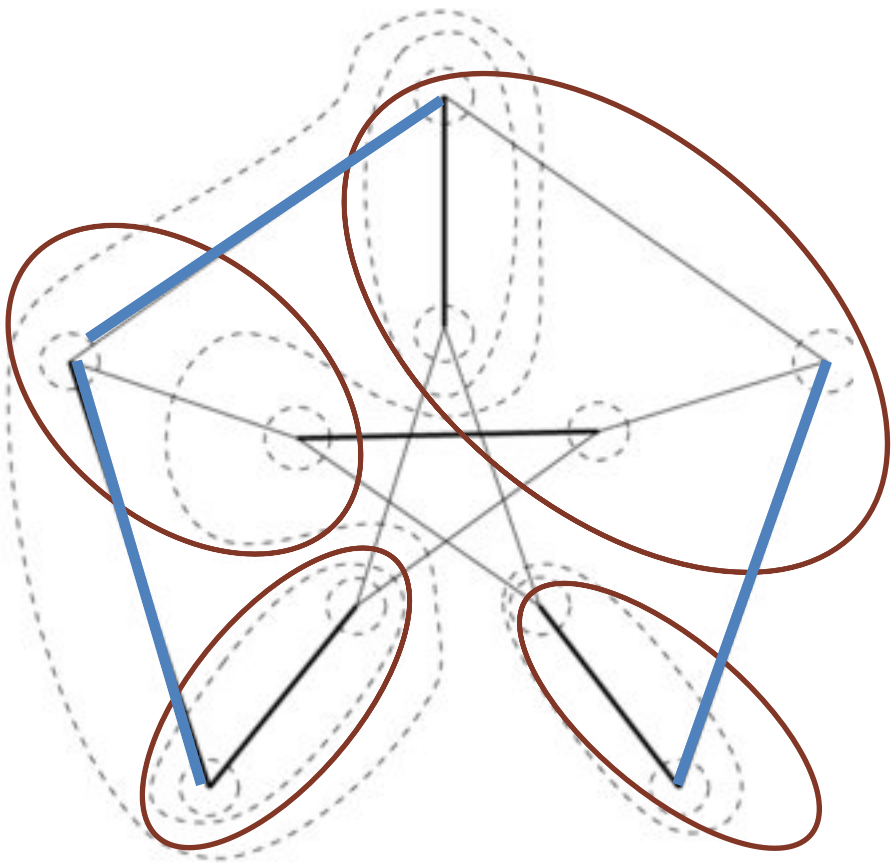
- مسئله:
  - ورودی: گراف
  - خروجی: جنگل فراگیر (با احتمال $1-1/n^b$)

# مسئله جنگل فراگیر



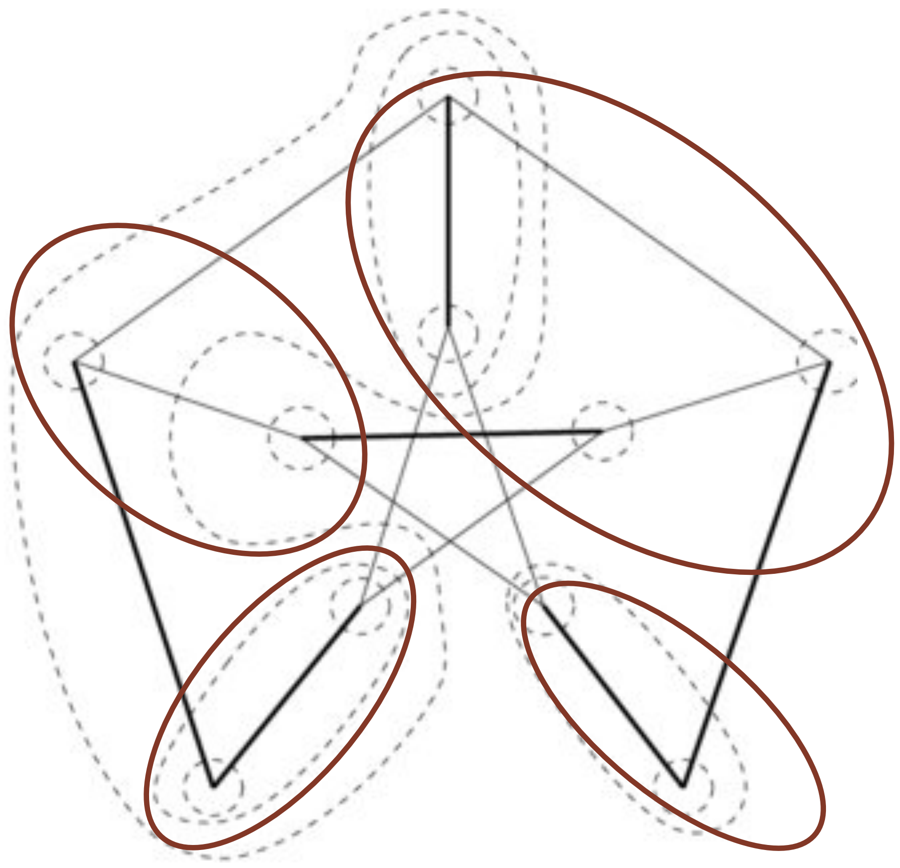
- مسئله:
  - ورودی: گراف
  - خروجی: جنگل فراگیر (با احتمال $1-1/n^b$)
- الگوریتم غیر جریان:
  - افرازی از راس‌ها
    - شروع: هر راس یک مجموعه از افراز
  - هر دفعه ($R=\log n$ بار):
    - یک یال خروجی برای هر مجموعه
    - اتصال دو مجموعه با یال‌ها

# مسئله جنگل فراگیر



- مسئله:
  - ورودی: گراف
  - خروجی: جنگل فراگیر (با احتمال $(1-1/n^b$
- الگوریتم غیر جریان:
  - افرازی از راس‌ها
    - شروع: هر راس یک مجموعه از افراز
  - هر دفعه ($R=\log n$ بار):
    - یک یال خروجی برای هر مجموعه
    - اتصال دو مجموعه با یال‌ها
- => هر دفعه تعداد دسته‌ها نصف می‌شود
  - در $\log n$ مرحله جنگل فراگیر

# مسئله جنگل فراگیر



- مسئله:
  - ورودی: گراف
  - خروجی: جنگل فراگیر (با احتمال $1-1/n^b$)

# روش خلاصه‌سازی AGM

- $x_u$: بردای همسایه بودن برای u (سطر u از ماتریس مجاورت)
  - اگر یال $u-v$ آنگاه $x_u[v] = \{$ اگر $u<v$ : 1 ؛ وگرنه $(v<u)$: 1- $\}$
  - وگرنه: $x_u[v] = 0$
- $x_A$: $\sum_{u\in A} x_u$

# روش خلاصه‌سازی AGM

- $x_u$: بردای همسایه بودن برای u (سطر u از ماتریس مجاورت)
  - اگر یال u–v آنگاه $x_u[v] = \{$ اگر u<v : 1 ؛ وگرنه (v<u): -1 $\}$
  - وگرنه: $x_u[v] = 0$
- $x_A$: $\sum_{u\in A} x_u$
- A زیرمجموعه راس‌ها، $support(\sum_{u\in A} x_u)$: یال‌های متصل به A

# الگوریتم جریانی برای جنگل فراگیر

- همیشه یک افراز از راس‌ها
  - برای هر مجموعه A:
    - $x_A$ یک SupportFind با k و $\delta < 1/(Rn^{1+b})$

# الگوریتم جریانی برای جنگل فراگیر

- همیشه یک افراز از راس‌ها
  - برای هر مجموعه A:
    - $x_A$ یک SupportFind با k و $\delta < 1/(Rn^{1+b})$
- برای R بار:
  - برای هر مجموعه A، یک یال را پیدا کن
  - برای هر یال: دو مجموعه A و B را ادغام کن
    - $x_A + x_B = x_{A \cup B}$
    - ادغام دو SupportFind

| | |
|---|---|
| 1 | $A_1$ |
| 2 | $A_2$ |
| … | … |
| log n | $A_{\log n}$ |

# تحلیل AGM:

- تعداد پرسش از SupportFindها: R مرحله، n مجموعه راس: Rn
  - احتمال خطا $< Rn\delta < 1/n^b$

# تحلیل AGM:

- تعداد پرسش از SupportFindها: R مرحله، n مجموعه راس: Rn
  - احتمال خطا < $Rn\delta$ < $1/n^b$
- حافظه: $O(nR\log^3 n)$
  - nR تا SupportFind

# تحلیل AGM:

- تعداد پرسش از SupportFindها: R مرحله، n مجموعه راس: Rn
  - احتمال خطا < Rn$\delta$ < $1/n^b$
- حافظه: $O(nR\log^3 n)$
  - nR تا SupportFind
- الگوریتم بهتر با $O(n\log^3 n)$ حافظه
  - SupportFind با دو نوع خطا
    - این دفعه نتوانستم (با احتمال زیاد)
    - خروجی غلط (با احتمال کم)