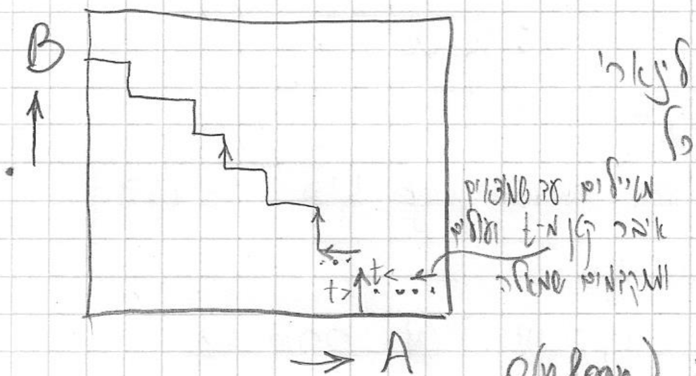
הרצאה (5)

עוד דוגמאות לאפליקציות של המטריצה המונוטונית...

<u>בעיה:</u> נתונות 2 קב' של מספרים ממשיים כ"א בגודל n, ופרמטר $1\le k\le n^2$.
<u>רוצים</u> למצוא את המספר ה-k בגודלו בקבוצה $A+B := \{x+y \mid x\in A, y\in B\}$.

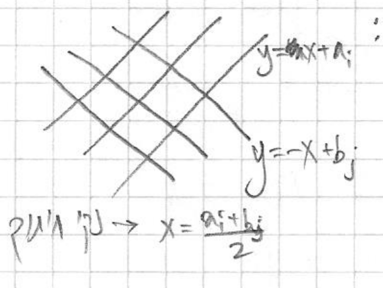
טבעי מאוד שנבקש לסדר את הסכומים כאיברים במטריצה. נסדר את איברי הקבוצות: $A=\{a_1<a_2<\dots<a_n\}$, $B=\{b_1<b_2<\dots<b_n\}$. וכעת המטריצה M: $M_{ij}=a_i+b_j$.
ורואים מיד ש-M מונוטונית.

<u>פרוצ' הכרעה:</u> בהינתן t, כמה מספרים ב-$A+B$ $\le t$?



מבקרים בגבול המתאר שמוכל במס' ליניארי של איברים. לא משנה אם מפעילים בכל שורה חיפוש בינארי, או שמקדמים איבר איבר.
לכן, $D(n)=O(n)$ $\Rightarrow$ סה"כ זמן ריצה $O(n\log n)$.

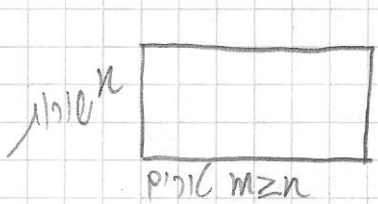
אפשר להסתכל על הבעיה כמקרה פרטי מיוחד של בעיית ה-Slope Selection כאשר G הישרים הם עם שיפוע 1 או שיפוע -1:



יש כמה סוגים של מטריצות מונוטוניות...

<u>חיפוש במטריצות מונוטוניות לגמרי</u> (totally monotone)

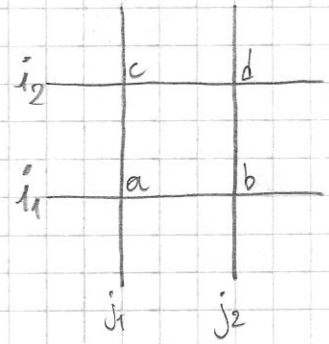
לא מניחים שהמטריצה ריבועית.



מטריצה מונוטונית מוגדרת כך: לכל שורה i נסמן ב- $J(i)$ את האינדקס של הטור עם האיבר המקס' בשורה.

מטריצה היא <u>מונוטונית</u> אם $\boxed{i_1<i_2 \Rightarrow J(i_1)\le J(i_2)}$
מטריצה היא מונוטונית לגמרי, אם כל תת מטריצה היא מונוטונית
$\equiv$ כל תת מטריצה של 2×2 מונוטונית

במילים פשוטות, ניקח 2 עמודות ו-2 שורות.



אסור שהמקס' ב-i_1 יתקבל ב-d והמקס'
ב-i_2 יתקבל ב-c.

≡ אם $a<b$ אז $c<d$.

2 מקרים פרטיים של העניין הזה:

<u>מטריצות Monge</u>

מט' Monge מקיימת את התכונה הבאה.
אם המט' היא M, ומסתכלים על תת מטריצה 2x2,
אז לכל $j>i$, $l>k$:

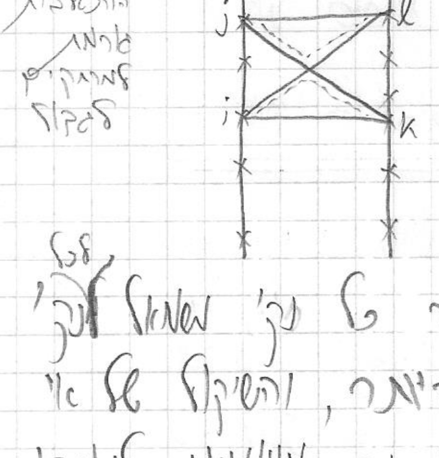
$$M_{ik}+M_{jl} < M_{il}+M_{jk}$$

המטריצה היא inverse Monge אם באותם תנאים מתקיים

$$M_{ik}+M_{jl} > M_{il}+M_{jk}$$

וזה גורר מט' מונוטונית לגמרי, $b+c<a+d$ (ואם $a<b$ אז $c<d$).

אגב, אם רוצים להפוך Monge ל-inverse-Monge, אז משאירים את העמודות איך שהן והופכים את סדר השורות. זה בעצם לא כ"כ חשוב מה מהם לצורך האלג'.



מונג' שם לב לקטע הבא:
אם המצב הוא כפי שמצויר משמאל אז

$$ik+jl < il+jk$$

וזה נובע בפשטות מאי-שוויון המשולש.

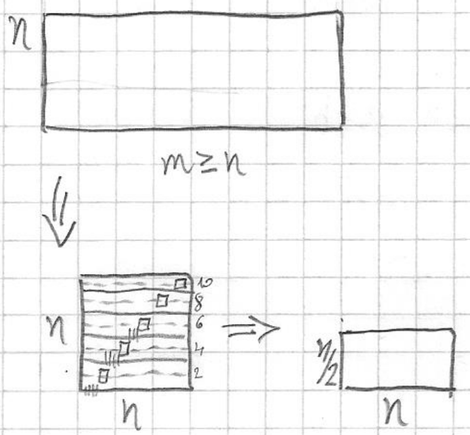
נניח שיש לנו מס' מכשולים במספרון, ורוצים לחבר כל נק' משמאל לכל נק' מימין, ורוצים לכל נק' את הדרך הקצרה ביותר, והשיקול של אי-שוויון המשולש עדיין נכון, ונקבל מטריצת Monge שהיא גם מונוטונית לגמרי!

כעת נציג את האלג' שפותר את הבעיה בזמן לינארי. איזו בעיה?
תכף נראה ←

הבעיה: למצוא את המקס' בכל שורה במט' מונוטונית לגמרי.
בזמן לינארי $O(n+m)$. האלג' הומצא ע"י חמישה של חברים שהראשי תיבות שלהם SMAWK. ('87)

Shor, Moran (טכניון), Aggarwal (לא הסלב), Wilber, Klawe

האלג' פשוט אך קצת טריקי:
כגדול, מה שנעשה זה לצמצם את המט' לריבועית ע"י סילוק טורים שלא יכולים להכיל מקס' בשום שורה (reduce)



מכאן, נוקים רק את השורות הזוגיות,
וקיבלנו שוב מט' מלבנית

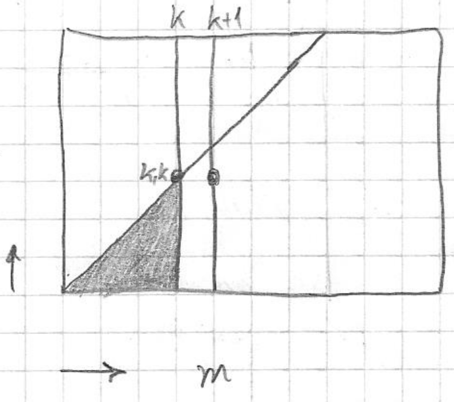
בשורות הזוגיות, נקבל את המקסימום,
ואז באי זוגית נוכל לנ"ל עד המקס' ורק בין המקס' של האי זוגית נצטרך לבדוק. בסה"כ מסתכלים על $O(n)$ איברים במטריצה, כלומר השלמה לשורות האי זוגית לוקחת זמן $O(n)$.

ובאופן קצת יותר מפורש:

```
ROWMAX(A,n,m)
  B := REDUCE(A)
  if n=1 then return   (מצאנו את המקסימום)
  else
    C := B של השורות הזוגיות
    ROWMAX(C, n/2, n)

    ונשלים את מציאת המקס' בשורות האי זוגיות
```

```
REDUCE(A,n,m)
C ← A , k ← 1
While C has more than n columns
במקרים:
(1) if C_{k,k} > C_{k,k+1} and k<n
        k ← k+1
(2) if C_{k,k} > C_{k,k+1} and k=n
        delete column k+1
(3) if C_{k,k} < C_{k,k+1} then
        delete column k, and k ← k-1
```

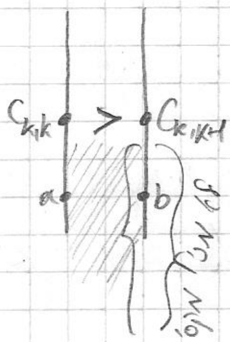


__אינווריאנטה:__

נסתכל על המשולש מהפינה השמאלית התחתונה, ושמה אלכסון (לא כולל) ועד הטור ה-k כולל:
שלא קיים בו מקס' של אף שורה.

בהתחלה האינווריאנטה נכונה באופן ריק.
אח"כ יש 3 מקרים:

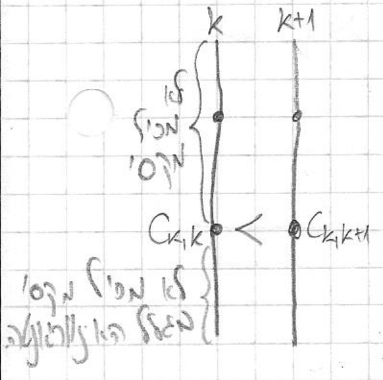
(1)



$\Rightarrow$ $b>a$ בכל החלק התחתון של 2 הטורים (אם $C_{k,k}>C_{k,k+1}$)

מכיון שבעמודה ה-$k+1$ אין שום מקס', ניתן להזיז את המשולש ימינה והאינווריאנטה נשמרת עבור $k+1$.

(2) אם הגענו למעלה, והתנאי מתקיים אז כל הטור ה-$k+1$ "מת" כלומר בכולו אין מקס' ואפשר למחוק אותו, והאינווריאנטה נשמרת.



(3) אותו שיקול כמו קודם, רק שכאן צריך להסתכל מעל האיברים, ולא יהיה שם מקס'.
גם למטה לא יהיה ולכן אפשר להזיז את הטור ה-k, ולא יודעים שום דבר על העמודה ה-k המקשה ולכן עושים $k \leftarrow k-1$.

במילים אחרות, שומרים על האינווריאנטה.

בסוף, המטריצה C היא ריבועית ולא מחקנו שום מקס'. זמן ריצה $O(n+m)$.

מקרים 2 ו-3 מתבצעים לכל היותר $n-m$ פעמים (כי נגמרים עם מטריצה ריבועית). ו-1 מתבצע $\leq (n-m)+m$ פעמים (כי מגדילים את k מ-1 ל-n...)

זמן הרדוקציה הוא $O(m+n)$ והיות ו- $n \le m$ זה בעצם $O(m)$.

כעת, זמן הרדוקציה של כל האלג': נחשב ע"י נוסחת רקורסיה שהיא יוצר:

$T(n,m) =$ הזמן למטריצה $n \times m$

$$T(n,m) = \underbrace{O(m)}_{\text{Reduce}} + T\left(\tfrac{n}{2}, n\right)$$

צמצום עמודות +
מילוי שורות
אי זוגיות

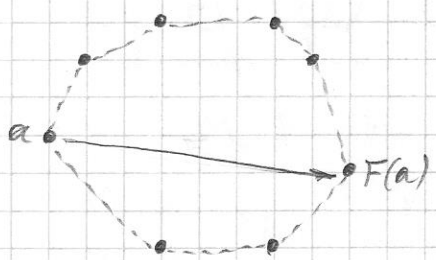
נשכנע שהפתרון אכן לינארי. ננחש $T(n,m) = An + Bm$

$$cm + A\tfrac{n}{2} + Bn \le An + Bm \rightarrow \underset{(\text{נבחר})}{B = C} \rightarrow \tfrac{A}{2} + B \le A \rightarrow \underset{(\text{נבחר})}{A = 2B = 2C}$$

והכל מסתדר עבור הבחירות שלנו, והשתכנענו.

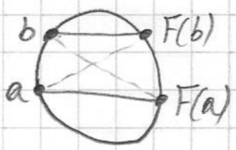
<u>דוגמה:</u>

נתון מצולע קמור בעל n קודקודים. רוצים למצוא לכל קודקוד את הקודקוד המרוחק ביותר ממנו.



במקרה זה, כל הקטעים $aF(a)$ חותכים זה את זה.

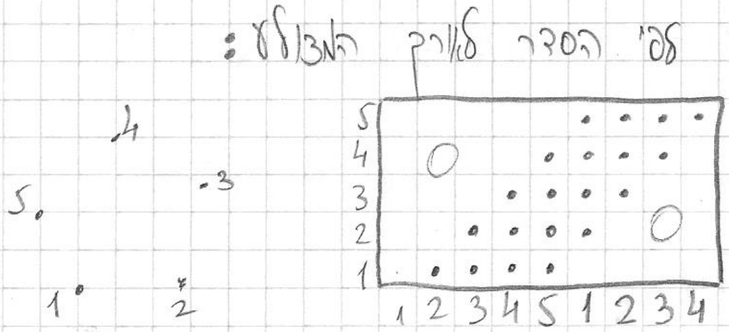
אם לא אז:



$$aF(a) + bF(b) < aF(b) + bF(a)$$

אבל זה לא ייתכן כי אז או ש- $aF(a) < aF(b)$ או ש- $bF(b) < bF(a)$ וזה לא ייתכן.

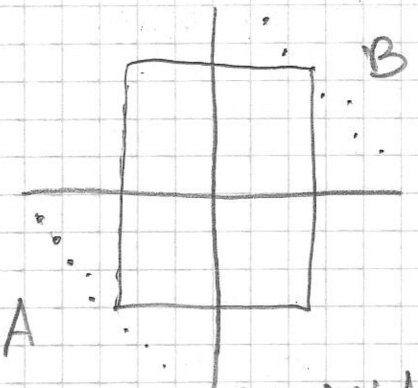
נסדר את הקודקודים במטריצה לפי הסדר לאורך המצולע:



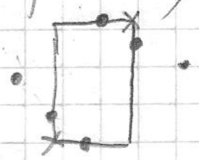
והאמת זה קצת מפתיע שאפשר למצוא את כל המידע הזה, בזמן לינארי, בלי שום לוגים.

עוד דוגמה;

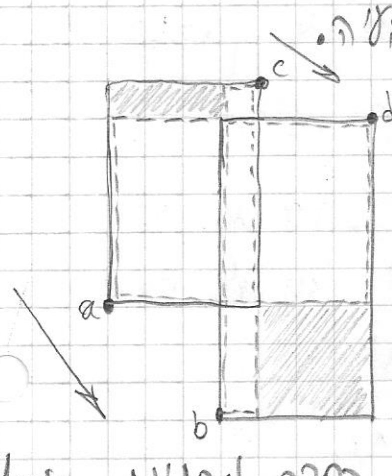
יש 2 קבוצות של נק'. האחת ברביע
השלישי והשניה ברביע הראשון.
והנק' הן גם מונוטונית יורדות ב- x,y.



נסתכל על מלבנים מקבילים לצירים עם קודקוד ב-A וקודקוד ב-B.
אנו מעוניינים בשטח המלבן המקס' ואפילו בשטח המלבן המקס' לכל A.
ואפשר שטח או היקף, אפילו יותר קל.
זה סוג של המלבן הריק המקסימלי. למעשה במקרה כזה נתעניין
במלבנים שיושבים על 2 נק' סמוכות מקב' אחת ו-2 נק' סמוכות
מקב' שניה.



אבל לכל זוג נק' מקב' אחת שעליהן יושב המלבן
ניתן לשים נק' אחת שעליה צריכה לשבת הפינה
ורק כך נקבל בדיוק את הניסוח הקודם של הבע"ה.



נצייר את 4 המלבנים בין a,b ו-c,d, ונקבל
ששטח המקווקוים יותר קטן כי הוא חלק משטח
של 2 האחרים.

נקבל: $S_{ac}+S_{bd} > S_{ad}+S_{bc}$

inverse Monje

אבל, במקרה של היקפים, יש שוויון, שזה גם בסדר מבחינת Monje

טוב, נמשיך. נעבור לנושא הבא, תכנות לינארי.

הסתכלנו על הנושא בקורס הבסיסי, ונחזור קצת על הרעיונות, ועוד.

תכנות לינארי

נתונים n אילוצים לינאריים ב-d משתנים, ונתונה פונק' מטרה לינארית. רוצים להביא את פונק' המטרה למינימום, תוך קיום כל האילוצים.

אם המשתנים שלנו הם: $(x_1,\dots,x_d)\in\mathbb{R}^d$

נסמן A מטריצה n×d, b וקטור באורך n. האילוצים הם: $Ax\le b$

ופונק' המטרה: $\min c^Tx$

ובאופן יותר מפורש:

$$a_{11}x_1+a_{12}x_2+\dots+a_{1d}x_d\le b_1$$
$$\vdots$$
$$a_{n1}x_1+a_{n2}x_2+\dots+a_{nd}x_d\le b_n$$

יש את אלג' הסימפלקס הידוע.

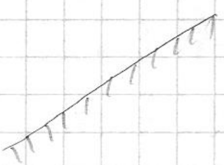
הבעיה הגדולה היתה האם הבעיה פולינומיאלית?

באופן חלש, התשובה היא כן! כלומר זמן הריצה הוא פולינומיאלי כפונק' של n,d ומספר הביטים שמקודדים את A,b,c.

לא ידוע האם קיים אלג' שזמן הריצה פולינומיאלי רק ב-d,n.

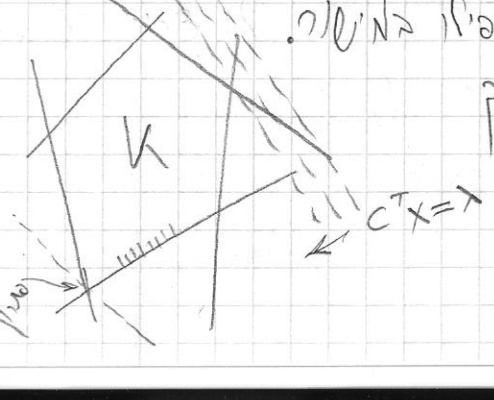
אלגוריתמים גאומטריים רצים בזמן $O(n)$ עם מקדם שגדל אקספ' כמו d. זה התחיל באלג' של מגידו ומאז היו קצת התפתחויות.

קצת על האינטרפרטציה הגאומטרית:



⊛ כל אילוץ הוא חצי מרחב התחום ע"י מישור.

⊛ K = האיזור הפיזיבילי, הוא חיתוך כל n החצאים המרחבים = פוליהדרון קמור, לאו דווקא חסום, אולי ריק, שיש לו ≤ n facets ‏(d-1)-מימדיות.



לא רוצים לחשב את k בצורה מפורשת, אפילו במישור.

נניח שיש k ונסתכל על על-מישור $c^Tx=\lambda$ מזיז כך ששינוי של λ, פורשים את λ במקום המתאים. (c^Tx = פונק' המטרה)

יתכן בעיה לא חסומה = K לא חסום בכיוון שבו $-\infty \downarrow x^T c$

ויתכן שהיא לא פיזיבילית, $K=\emptyset$

אחרת, המינימום מושג בקדקד.

סיבוכיות K - מס' הקדקדים של פאון קמור עם n facets היא $O(n^{\lfloor d/2 \rfloor})$.

שזה the upper bound theorem.

אפילו במישור, לחשב את K יעלה $O(n\log n)$, ואנחנו כאמור רוצים להמנע מלחשב את K באופן מפורש. אנו מעוניינים באלג' שמבקר במעט קדקדים של K, וזה לינארי ב-n ומאוד לא יעיל ב-d.

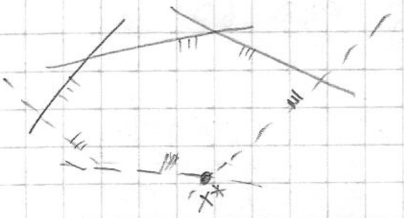
תזכורת: <u>אלג' של מג'ידו במישור</u> $O(n)$

הא' שיוויונים שלנו הם $y \le a_i x + b_i$

$y \ge a_i x + b_i$

נג'יד שיש גם אילוצים: $c \le x \le d$ שהם לא באמת מעניינים.

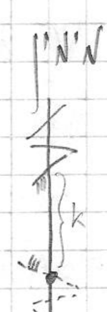
ופונק' המטרה שלנו היא $\min y$.



הרעיון של מג'ידו: המינימום מושג באיפשהו x^*.

פונק' ההכרעה כאן, מה שמג'ידו הגדיר כ- oracle, היא בהנתן x_0: האם $x_0 \gtreqless x^*$.

איך עובד ה- oracle? רוצים להכריע האם x^* נמצא מימין או משמאל (או ב-) ל-x_0.

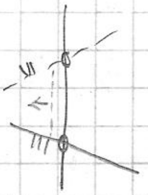


קודם כל עוברים על כל האילוצים מהצורה $y \ge a_i x + b_i$ ומחשבים את המקס' שלהם

כנ"ל כל האילוצים $y \le a_i x + b_i$ ומחשבים מינימום ב-x_0.

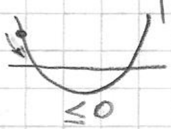
אם ה- min מעל המקס', אז החיתוך לא ריק.

אם החיתוך לא ריק, כלומר $x_0 = x$ חותך את K, כל מה שצריך כאן לעשות זה לבדוק את הגזירה בחיתוך התחתון (המקס') ולראות לאן היא משתפלת - שם המינימום, כי K קמור.



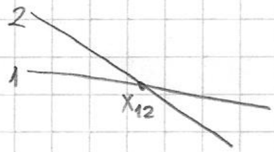
אם $X=X_0$, לא נוחק את K, המצב הוא קצת יותר מאתגר, וצריך להכריע באיזה צד X^*.

במקרה כזה, מסתכלים על 2 האילוצים, ובודקים באיזה צד, המרחק ביניהם מתקצר. כך, יודעים ש-K יהיה חייב להיות באותו צד, ושם יהיה X^*.



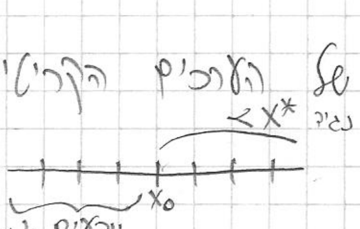
אם המרחק ביניהם גדל גם ימינה וגם שמאלה, אז יודעים שאין חיתוך כלומר $K=\emptyset$.
כל חישובי המקס' והמינ' לוקחים זמן לינארי.

כעת מיקבול:
מה האלג' עושה? קודם כל מחשב מקס' ומינ' של אילוצים. חישוב מקס' יכול להתבצע בזוגות, בעומק של $\log n$.
מגידו עשה משהו יותר טוב.
נסתכל על הצעד הראשון של המימוש המקבילי של האלג': מציאת מינימום/מקסימום ע"י בדיקת זוגות של ישרים.

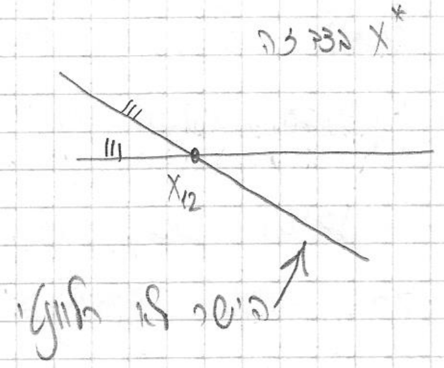


כדי לדעת מי מהישרים 1 או 2 יותר גבוה ב-X^*, צריך להשוות את X^* עם X_{12}.

והרי זה מה שעושה החיפוש הפרמטרי. הוא אוסף שאילתות רבות, עונה על $\log n$ מהן עם n מעבדים... למעשה עושים חיפוש בינארי עליהם.
הצעד הראשון: מבצע $\frac{n}{2}$ השוואות ← ערכים קריטיים (X_{12}) ומבצעים חיפוש בינארי.
הצעד הראשון של החיפוש: מחשב חציון X_0 של הערכים הקריטיים וקורא לאורקל בו $X^*:X_0$.



אפשר לעצור בנק' זו, ולזרוק $\frac{n}{4}$ מהישרים.
נזרוק אותם ונתחיל מההתחלה.



אילו ישרים זורקים?
כל אלו שיודעים את התשובה,
זורקים את הישר כדי שמשמאל
כי האופט' לא יהיה עליו.

זהו אלג' דטרמיניסטי שרץ בזמן לינארי: $O(n)$.
מגידו הכליל את העניין לכל ממד בממדים גבוהים. כל ההכללות רצות בזמן nC_d כאשר C_d גדל אקספוננציאלית ב-d, וגם האלג' מאוד מסובכים.

כיוון אחר שנחקר הוא גישה רנדומית, בה כל פעם מוסיפים את האילוצים בסדר רנדומי, והאופטימום הפיזיבילי הולך ומשתנה, כאשר כל פעם צריך (אולי) לתקן את הפתרון עד שבסוף יהיה קדקד של A.