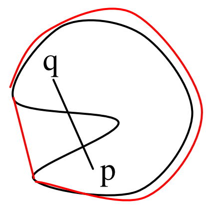
## אלגוריתמים גיאומטריים

### חישוב הקמור של קבוצה סופית של נקודות במישור.

הגדרה: קבוצה S של נקודות (במישור) היא קבוצה קמורה (convex), אם לכל זוג נקודות $p,q \in S$, הקטע $\overline{pq}$ מוכל ב $S$.

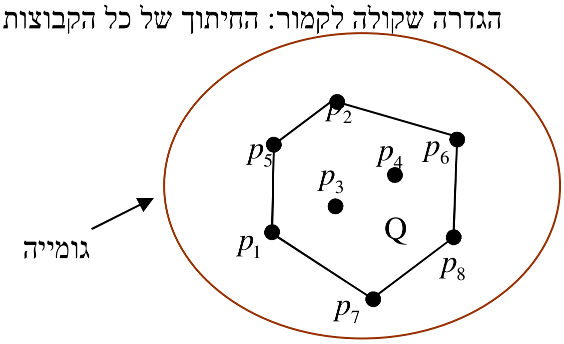
לדוגמה, מעגל הוא קבוצה קמורה, כי כל שתי נקודות בו – הקטע ביניהן מוכל במעגל.




אבל: הוא לא צורה קמורה .

הקמור (Convex Hull) של קבוצה S היא הקבוצה הקמורה המינימאלית (ביחס להכלה) שמכילה את S.
(בדוגמה – עם הקו האדום).

הגדרה שקולה לקמור: החיתוך של כל הקבוצות הקמורות שמכילות את S.




אבחנה

בהינתן קבוצה סופית של נקודות במישור Q, $CH(Q)$ יהיה המצולע הקמור הקטן ביותר שמכיל את Q ובקודקודיו יהיו תת קבוצה של Q.

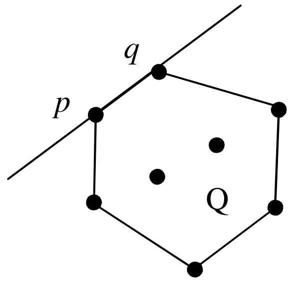
אינטואיטיבית, זו הצורה שנקבל אם נתקע מסמר בכל נקודה ב Q ונשחרר גומייה מסביב לכל הנקודות בQ.

בעיה

בהינתן קבוצה Q של n נקודות במישור, לחשב את CH(Q), כלומר, מי הם הקודקודים של CH(Q) ומה הסדר שלהם (לפי כיוון השעון).

בדוגמה: $p_1, p_5, p_2, p_6, p_8, p_7$ .

אם נסתכל על צלע $pq$ של הקמור כך ש $p$ קודמת ל $q$ בסדר ונעביר ישיר דרך $\overline{pq}$ שכיוונו $\overrightarrow{pq}$ אז כל הנקודות נמצאות מימין או על הישר ולהיפך: אם עבור זוג נקודות $p,q \in Q$ הישר העובר דרך $\overline{pq}$ וכיוונו $\overrightarrow{pq}$ מכיל את כל הנקודות $Q \setminus \{p,q\}$ מימינו, אז $pq$ היא צלע של הקמור.

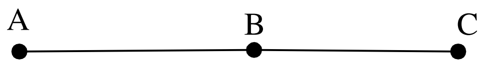



לפי עובדה זאת, ניתן להציע אלגוריתם (נאיבי) לחישוב הקמור.

- לכל זוג סדור של נקודות, נבדוק האם כל שאר הנקודות נמצא מימין לישר המוגדר ע"י זוג הנקודות. אם כן, אז שתי הנקודות הן קודקודים של הקמור. הסיבוכיות היא $O\left(n^3\right)$
- חישוב הסדר של הקמור $O\left(n^2\right)$.

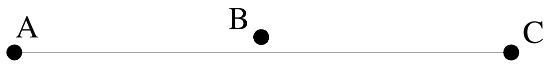
סה"כ: $O\left(n^3\right)$

<u>בעיה:</u> אם יש לנו 3 נקודות $A,B,C$ על ישר אחד כך שכל שאר הנקודות מימינו, אז נקבל תשובה חיובית עבור $AB,BC,AC$



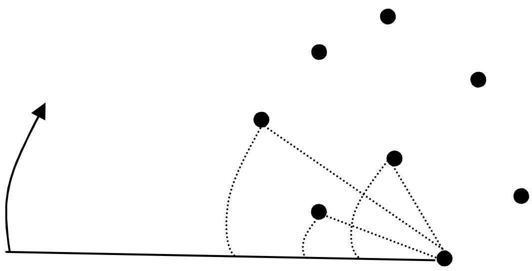

<u>פתרון:</u> נבדוק האם כל הנקודות נמצאות מימין לישר המוגדר ע"י זוג הנקודות הנוכחי או <u>על הקטע הביניהן</u>.

<u>בעיה יותר חמורה:</u> נניח שיש לנו 3 נקודות, לא על ישר אחד, אלא כמעט על ישר אחד.




הפתרון הנכון הוא משולש, אולם בגלל טעות חישוב בגלל בעיות של דיוק, עלולים שלא לקבל כלל <u>מצולע</u>. $\Leftarrow$ האלגוריתם לא יציב (robust) לטעויות חישוב.

אלגוריתם עטיפת מתנה – gift wrapping (Jarviss scan).



פורמאלית:

תהא $p_0$ הנקודה הנמוכה ביותר ב Q (אם יש כמה כאלה , נבחר את הנקודה השמאלית ביותר).

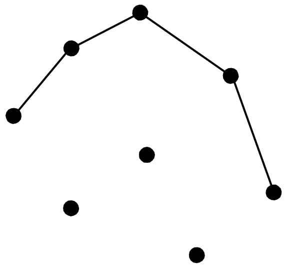
הנקודה הבאה בקמור היא הנקודה $p_i$ כך שהזווית בין הכיוון השלילי של ציר ה X והווקטור $\overrightarrow{p_0p_i}$ היא מינימאלית (ואם יש כמה נקודות כאלה, בוחרים את זו שהכי רחוקה מ $p_0$ ).

תהא $p_i$ הנקודה האחרונה שהצטרפה לקמור ותהא $p_j$ הנקודה הקודמת לה בקמור.

הנקודה בקמור $p_k$ היא זו שיוצרת זווית מינימאלית בין הקרן שמתחילה ב $p_i$ וכיוונה $\overrightarrow{p_i p_j}$ לבין הווקטור $\overrightarrow{p_i p_k}$ (אם יש כמה כאלה נבחר את זו שהכי רחוקה מ $p_i$).

האלגוריתם יציב לטעויות חישוב, כלומר מבטיח שנקבל מצולע חוקי (לאו דווקא הקמור).
<u>סיבוכיות</u> $O\left(n^2\right)$ או יותר מדוייק: $O\left(nh\right)$ כאשר $h$ הוא מספר הנקודות בקמור.
$output-consitive$

האלגוריתם הבא מחשב את החלק העליון של הקמור (Upper Convex Hull – UCH)
אפשר בצורה דומה לחשב את החלק התחתון ולשרשר את התוצאות.



- בשלב הראשון ממינים את הנקודות בסדר לקסיקוגרפי:
$\left(\left(x_1 = x_2 \;\&\; y_1 < y_2\right) \mid x_1 < x_2\right) \Leftrightarrow \left(x_1, y_1\right) < \left(x_2, y_2\right)$
- בשלב השני מוסיפים את הנקודות לפי הסדר ודואגים לעדכן את ה UCH של כל הנקודות שהוספנו (אלגוריתם אינקרמנטלי).

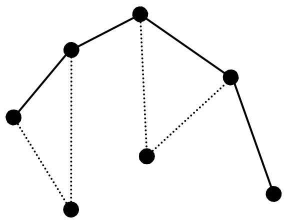
האלגוריתם מבוסס על העובדה הבאה: כאשר "מטיילים" עם כיוון השעון על מצולע קמור אז במעבר מצלע לצלע פונים ימינה. ולהיפך: כל מצולע שמקיים תכונה זו הוא קמור.

האלגוריתם Upper Convex Hall (Graham's walk/scan):
1. ממיינים את הנקודות בסדר לקסיקוגרפי.
2. דוחפים למחסנית S את שתי הנקודות הראשונות בסדר.
3. תהא $p_i$ הנקודה הבאה בסדר. דוחפים את $p_i$ ל S.
4. אם $|S|<3$ חזור ל 3.
5. אחרת, יהיו $p_j$ ו $p_k$ הנקודות שקודמות ל $p_i$ במחסנית, כלומר המחסנית נראית כך:

| |
|---|
| $p_i$ |
| $p_j$ |
| $p_k$ |
| |

(המחסנית גדלה למעלה)
6. אם הפניה מ $\overrightarrow{p_k p_j}$ אל $\overrightarrow{p_j p_i}$ היא לא פניה ימינה, זרוק את $p_j$ מהמחסנית וחזור ל 4.
7. אחרת, חזור ל 3.

דוגמה:



<u>סיבוכיות:</u>

- מיון: $O(n\log n)$
- עדכון לאחר הוספת נקודה $O(n)$ יתבצע $O(n)$ פעמים.
- סה"כ לכאורה $O(n^2)$

<u>ניתוח יותר אינטליגנטי:</u>

סיבוכיות כל העדכונים:

- מוספים נקודה: $O(n)$
- בדיקת פניה ימינה שמצליחה $O(n)$
- בדיקת פניה ימינה שנכשלת $\Leftarrow$ מחיקת נקודה $O(n)$.

$\Leftarrow$ סה"כ $O(n)$, כלומר סיבוכיות האלגוריתם מיון + $O(n)$, זאת אומרת $O(n\log n)$.

<u>נכונות:</u>

האלגוריתם מבטיח שנקבל שרשרת של פניות ימינה.

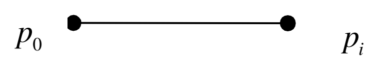
נוכל להראות שלא יתקבל מצב שבו נקודה נמצאת מעל ה UCH.

נוכיח באינדוקציה על סדר הוספת הנקודות:

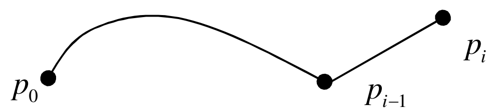
לאחר הוספת $p_i$ כל הנקודות $p_0,\dots,p_i$ נמצאות על או מתחת ל UCH הנוכחי.

<u>בסיס:</u>

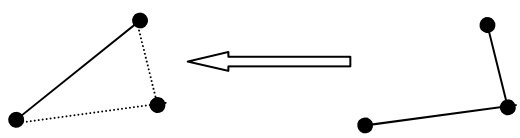
מתקיים!




<u>צעד:</u> לפני הוספת $p_i$ לפי הנחת האינדוקציה, כל הנקודות $p_0,\dots,p_{k-1}$ נמצאות על או מתחת ל UCH הנוכחי (אבל לא מעליו). כאשר מוסיפים את $p_i$ היא האחרונה בשרשרת, כלומר- היא נמצאת על UCH ובינה ובין $p_{i-1}$ אין אף נקודה אחרת ולכן התכונה נשמרת.




מספיק להראות שלאחר כל פעולות העדכון התכונה נשמרת, וזה אכן קורה כי בכל פעולת עדכון "מקצרים" פניות שמאלה וכך הנקודה האמצעית מבין שלוש הנקודות שמגדירות את הפניה יורדת מתחת ל UCH:



ולכן התכונה נשמרת ולאחר כל פעולות העדכון – האלגוריתם נשאר יציב.

<u>טענה:</u> לא ניתן לחשב את הקמור של קבוצה סופית של נקודות במישור בזמן טוב יותר מ $O(n\log n)$ במקרה הגרוע ביותר.

<u>הרדוקציה:</u> $x_i \to (x_i, x_i^2)$ כיוון שהפונקציה קמורה – הקמור שלהן יהיה סדר ה $x$-ים הממוין.

לכן מיינו בסיבוכיות נמוכה מ $O(n\log n)$ ולכן יש סתירה, כיוון שהוכחנו במבני נתונים שלא ניתן למיין בסיבוכיות נמוכה מ $O(n\log n)$.