ברוכים הבאים לתרגול 2 ☺

שחר אנגל

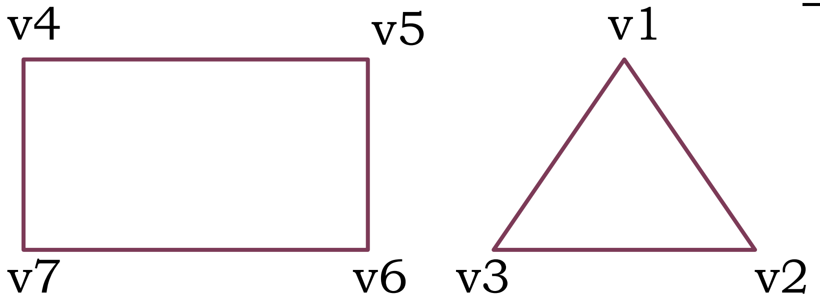
shaharbel0@gmail.com

תרגול- ימי שני 14-16 וימי חמישי 13-15

נושאי התרגול

1. מציאת מטריצת מסלולים - FW
2. האם יש מסלול בין vi ל-vj?
3. האם הגרף קשיר?
4. כמה רכיבי קשירות יש?
5. מיהם הקודקודים ברכיבי הקשירות?
6. מציאת מטריצת מסלולים
7. בקבוקים- האם קיים מסלול בין (i,j) ל-(k,l)?
8. אם כן, תנו דוגמא
9. סידור מטריצה
10. בעיית השוקולד

- בתרגול שעבר יצרנו מטריצת שכנויות. הגדרנו אותה כך- בכל תא שמופיע בו 1 או true נדע שיש קשר (או קשת) בין המצבים, ובתאים שמופיע בהם 0 או false אין קשר.
- בהינתן מטריצת שכנות, איך אפשר לדעת האם קיים מסלול, ואם כן, אז מהו?
- <u>לדוגמא:</u>

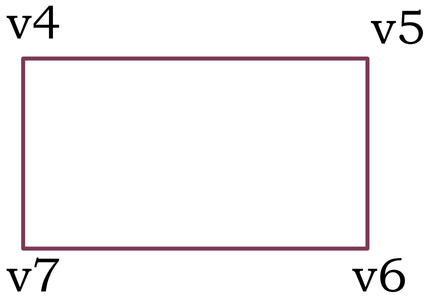


	v1	v2	v3	v4	v5	v6	v7
v1		1	1				
v2	1		1				
v3	1	1					
v4					1		1
v5				1		1	
v6					1		1
v7				1		1	

- האם יש צלע בין v1 ל-v2?
  - נלך לתא המתאים בטבלה- אם יש שם 1 אז יש צלע ואם יש שם 0 אז אין צלע
- האם יש מסלול בין v4 ל-v6?
  - בשביל מסלול הטבלה לא עוזרת לנו ישירות, כי אולי אין צלע, אבל יכול להיות שיש מישהו שמחבר ביניהם ויוצר מסלול

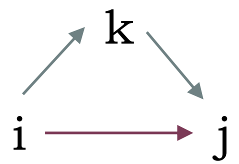
1. אלגוריתם Floyd-Warshall

- הרעיון שעומד מאחורי האלגוריתם הזה הוא לבדוק האם קיים קודקוד כלשהו בגרף שיכול לחבר לנו בין 2 קודקודים אחרים וליצור מסלול.
- <u>לדוגמא:</u> בשביל לענות על השאלה מקודם 'האם יש מסלול בין v4 ל-v6?' נוכל לראות שקודקוד v7 יכול לחבר ביניהם ולגרום לכך שיהיה מסלול.



- כלומר, האלגוריתם עובר על כל הקודקודים ובודק:

אם קיים מסלול בין i ל-k ובין k ל-j אז קיים מסלול בין i ל-j



- <u>האלגוריתם:</u>

```
for k=1 to n:
    for i=1 to n:
        for j=1 to n:
            T[i,j] = T[i,k]^T[k,j]
```

- מה הבעיה באלגוריתם?
  - שאם יש לי כבר מסלול בין i ל-j, אז k יכול להפריע לי- כי אולי אין צלע בין i ל-k או בין k ל-j.

1. אלגוריתם Floyd-Warshall

- מה הפתרון?
  - להוסיף תנאי 'או' שלא יפגע במה שקיים בתא, כלומר קודקוד לא יכול לפגוע אלא רק לעזור

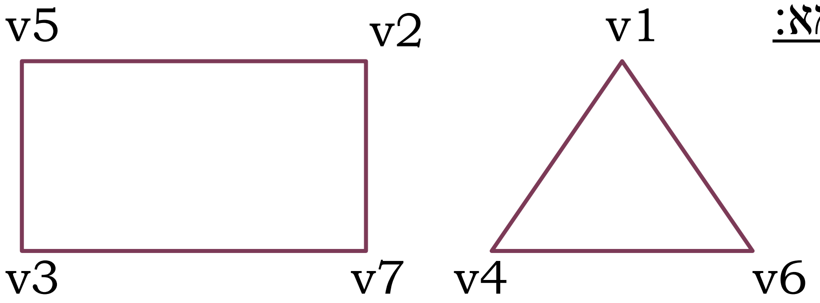
```
for k=1 to n:
    for i=1 to n:
        for j=1 to n:
            T[i,j] = (T[i,k] ^ T[k,j]) v T[i,j]
```

- ברגע שנריץ את האלגוריתם על מטריצת השכנויות שלנו, נוכל לדעת האם קיים מסלול בין כל 2 קודקודים.
- האם יש מסלול בין קודקוד לעצמו?
  - תשובה: כן, למשל המסלול של קודקוד v1 לעצמו הוא- v1-v2-v3-v1.
  - או, שניתן אפילו לומר שיש מסלול באורך 0.
  - **חידוד**: v1 לא שכן של עצמו, אבל קיים מסלול ממנו אל עצמו.
- בואו נבצע הרצה של FW ונקבל את מטריצת המסלולים:

	v1	v2	v3	v4	v5	v6	v7
v1	1	1	1				
v2	1	1	1				
v3	1	1	1				
v4				1	1	1	1
v5				1	1	1	1
v6				1	1	1	1
v7				1	1	1	1

1. אלגוריתם Floyd-Warshall

- ניקח בחשבון שלא תמיד המטריצה תהיה כזו יפה.
- <u>לדוגמא:</u>



	v1	v2	v3	v4	v5	v6	v7
v1	1			1		1	
v2		1	1		1		1
v3		1	1		1		1
v4	1			1		1	
v5		1	1		1		1
v6	1			1		1	
v7		1	1		1		1

- המטריצות נותנות לנו את אותו המידע, רק בסדר קודקודים שונה..
- האם ניתן להגיע ממטריצה מבולגנת כזו למטריצה מסודרת כמו מקודם?
  - כן, ע"י החלפת סדר הקודקודים כמו שהיה מקודם- נחליף עמודות ושורות.
  - איך עושים את זה בצורה יעילה? נסו לחשוב לבד..

2. **האם יש מסלול בין vi ל-vj?**

- נפעיל FW על הגרף ואז ב-o(1) נוכל לגשת לתא המתאים במטריצה ולראות האם יש שם 1 או 0

3. **האם הגרף קשיר? (אנו מדברים רק על גרפים לא מכוונים)**

- ברגע שכל המטריצה מלאה באחדות נדע שהגרף קשיר כי ניתן להגיע מכל קודקוד לכל קודקוד אחר.

  ב-o(n^2) נוכל לעבור על כל המטריצה ולבדוק

- האם יש פתרון יעיל יותר?
  - ב-o(n) נעבור על השורה הראשונה ונבדוק שהכל שם אחדות.
  - למה זה נותן פתרון?

4. **כמה רכיבי קשירות יש? (אנו מדברים רק על גרפים לא מכוונים)**

- נעבור על המטריצה ונבדוק.
- מה הבעיה?
  - שלא דווקא נקבל מטריצה כזו יפה
- אז מה נעשה?

	v1	v2	v3	v4	v5	v6	v7
v1	1	1	1				
v2	1	1	1				
v3	1	1	1				
v4				1	1	1	1
v5				1	1	1	1
v6				1	1	1	1
v7				1	1	1	1

4. כמה רכיבי קשירות יש? (אנו מדברים רק על גרפים לא מכוונים)

- נייצר מערך עזר

v1	v2	v3	v4	v5	v6	v7
0	0	0	0	0	0	0

- נעבור על המערך וכל פעם שנראה 0 נדע שעוד לא הגענו לקודקוד הזה. נבדוק במטריצת השכנויות מי השכנים שלו ונסמן במערך. נוסיף counter שבודק לנו מה מס' רכיבי הקשירות ונעלה אותו כל פעם. כך נעבור על כל המערך, ורק על השורות הרלוונטיות במטריצה.

- <u>דוגמת הרצה:</u>

	v1	v2	v3	v4	v5	v6	v7
v1	1			1		1	
v2		1	1		1		1
v3		1	1		1		1
v4	1			1		1	
v5		1	1		1		1
v6	1			1		1	
v7		1	1		1		1

v1	v2	v3	v4	v5	v6	v7

counter =

4. כמה רכיבי קשירות יש? (אנו מדברים רק על גרפים לא מכוונים)

- כך קיבלנו מערך המכיל לנו מידע עבור מספר רכיבי הקשירות

5. מיהם הקודקודים ברכיבי הקשירות?

- נוכל לעבור על המערך ולראות מי נמצא באיזה רכיב

- <u>שאלת מחשבה:</u> איך אפשר למצוא איבר שלא נמצא בתת קבוצה מסוימת? כלומר, אם יש לי 7 קודקודים וקבוצה $\{2,3,5,7\}$ איך אוכל בצורה יעילה לדעת מי לא נמצא שם? במקום לעבור על המערך ולראות איפה נשארו אפסים..

6. מציאת מטריצת מסלולים

- מקודם מצאנו מטריצת שכנויות שמראה לנו בין אלו קודקודים יש מסלולים, אך איך נדע מהם המסלולים?
- נייצר מטריצת עזר של מחרוזות ובכל תא נרשום איך מגיעים מקודקוד לקודקוד. זה לאו דווקא המסלול הקצר ביותר, אבל לפחות נותן לנו פתרון.
- ניקח את FW ונמיר את התנאי האחרון בו לתנאי שיעזור לנו לשרשר מסלולים.

7. **בקבוקים- האם קיים מסלול בין (i,j) ל-(k,l)?**

- נחזור לבעיית הבקבוקים. עכשיו, אם נרצה לדעת האם קיים מסלול בין מצב מסוים למצב אחר נוכל להמיר מ-(i,j) ל-k, להפעיל FW להמיר חזרה מ-k ל-i ו-j ולראות האם קיים מסלול או לא. אם ממש נרצה את המסלול נפעיל את האלגוריתם שמחשב לנו את המסלולים עצמם ונחזיר את הפתרון.

8. **אם כן, תנו דוגמא**

- לאחר שתריצו את 7 החזירו דוגמא ספציפית יותר.

9. **סידור המטריצה**

- דיברנו על זה מקודם. נחליף עמודות ושורות ונהפוך את המטריצה למטריצה יפה יותר

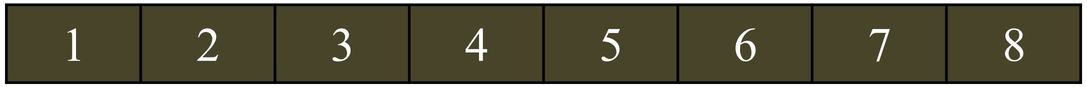
תיאור הבעיה

- **בעיית השוקולד**

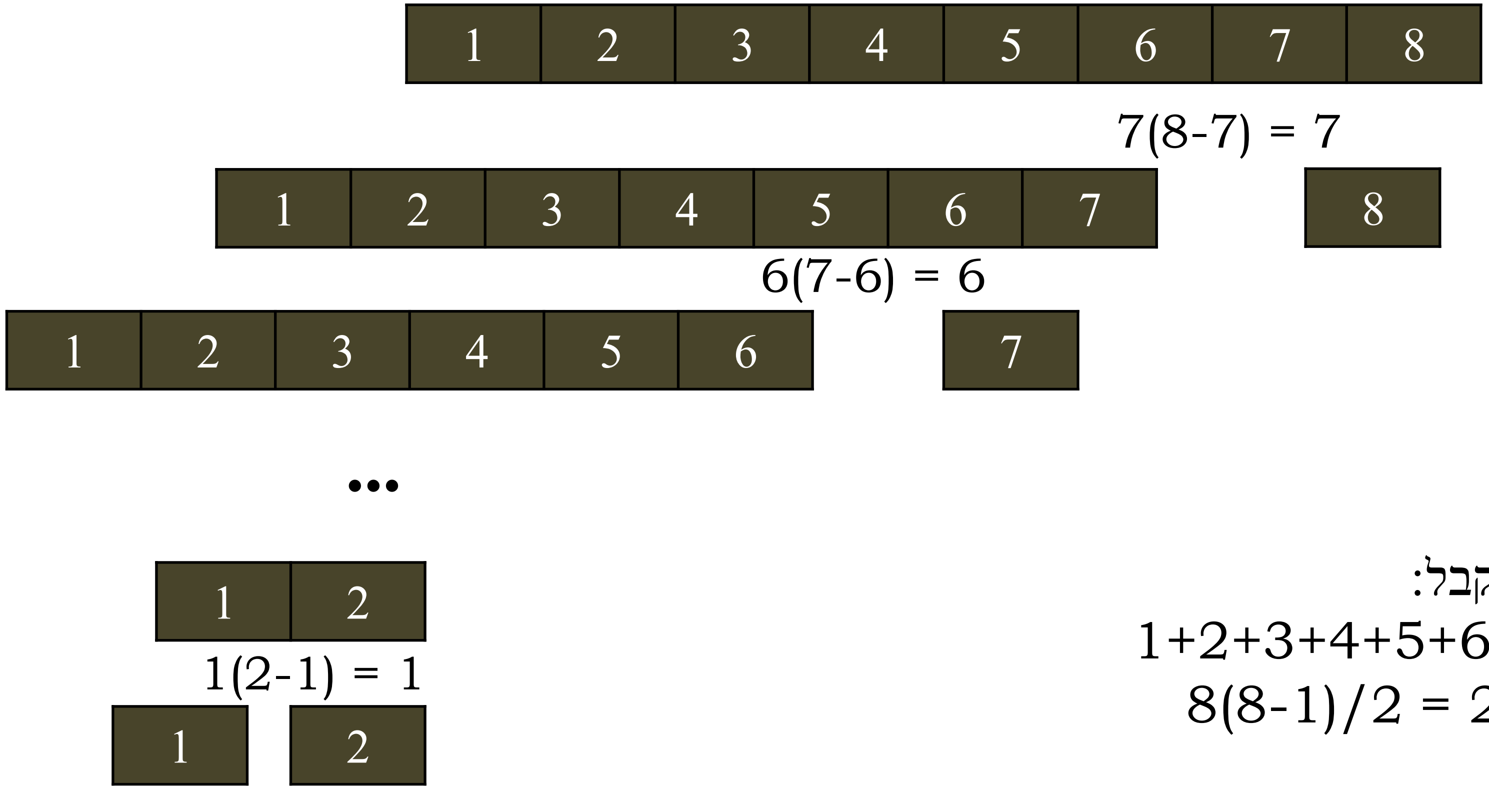
- יש פס שוקולד המכיל n קוביות. על כל חלוקה באינדקס k משלמים k(n-k)
- המטרה: לחלק את השוקולד לקוביות בודדות ולשלם את המינימום האפשרי.

10. בעיית השוקולד

- <u>לדוגמא:</u> ניקח פס שוקולד המכיל 8 קוביות:



- ניסיון 1: שבירה לאחדות



סה"כ נקבל:

$1+2+3+4+5+6+7=$

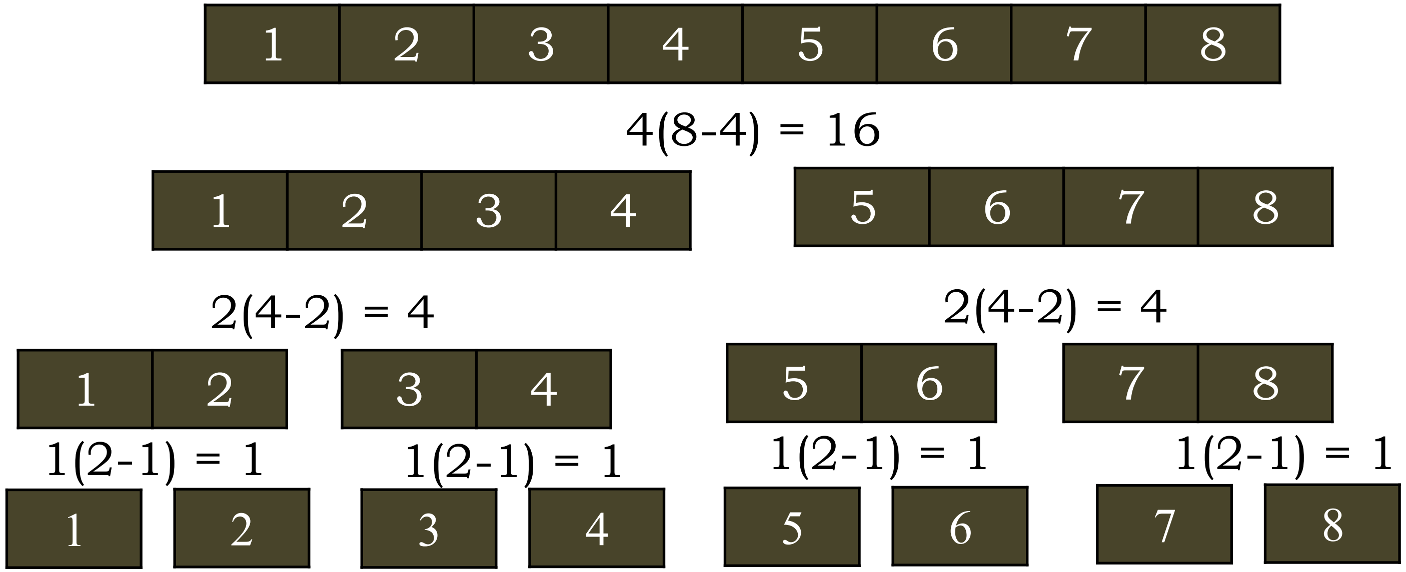
$8(8-1)/2 = 28$

10. בעיית השוקולד

- <u>לדוגמא:</u> ניקח פס שוקולד המכיל 8 קוביות:

1	2	3	4	5	6	7	8

- ניסיון 2: שבירה באמצע

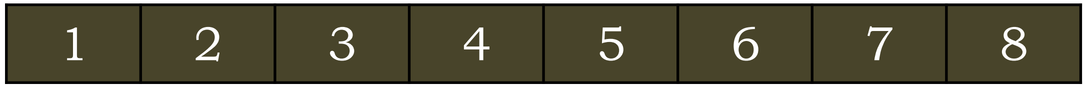


סה"כ נקבל:

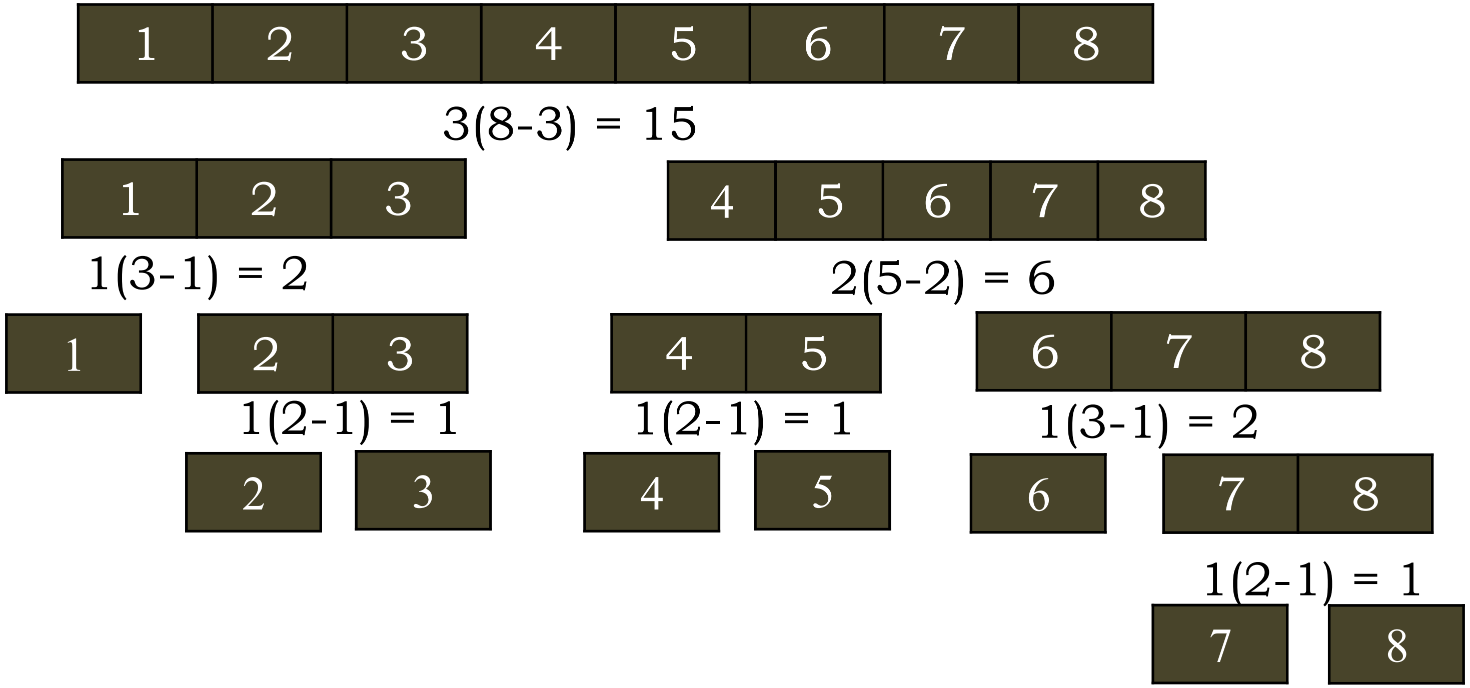
16+8+4=28

10. בעיית השוקולד

- לדוגמא: ניקח פס שוקולד המכיל 8 קוביות:



- ניסיון 3: פיבונאצ'י



סה"כ נקבל:

$15+2+6+1+1+2+1=28$

10. בעיית השוקולד

- לדוגמא: ניקח פס שוקולד המכיל 8 קוביות:

1	2	3	4	5	6	7	8

- מסקנה: על כל חלוקה אפשרית נצטרך לשלם $n(n-1)/2$
- **הוכחה:** נוכיח באינדוקציה על מספר הקוביות
- בסיס: $n=2$. אם יש לנו רק 2 קוביות יש רק פיצול אחד שנשלם עליו $1 = 2(2-1)/2=1(2-1)$
- הנחה: הטענה נכונה עבור $k \leq n-1$
- שלב האינדוקציה: נוכיח עבור $k=n$
- כאשר נפצל ב-k כלשהו, כל תת מקטע יהיה בגודל קטן או שווה ל-$n-1$. החלוקה עצמה תעלה לנו $k(n-k)$.
- מקטע שמאל $[1..k]$ יעלה לנו $k(k-1)/2$ לפי הנחת האינדוקציה
- מקטע ימין $[k+1...n]$ יעלה לנו $(n-k)(n-k-1)/2$ לפי הנחת האינדוקציה
- כלומר העלות הכוללת תהיה:

$$k(n-k)+\frac{k(k-1)}{2}+\frac{(n-k)(n-k-1)}{2}$$

10. בעיית השוקולד

- לדוגמא: ניקח פס שוקולד המכיל 8 קוביות:

1	2	3	4	5	6	7	8

- נמשיך לפתור:

$$\frac{2kn-2k^2+k^2-k+n^2-nk-n-kn+k^2+k}{2}$$

$$\frac{\cancel{2kn}\cancel{-2k^2}+\cancel{k^2}\cancel{-k}+n^2\cancel{-nk}-n\cancel{-kn}+\cancel{k^2}\cancel{+k}}{2}$$

$$\frac{n^2-n}{2} \Rightarrow \frac{n(n-1)}{2}$$

- ונגיע למסקנה ממקודם- לא משנה איזה חלוקה נבצע תמיד נשלם n(n-1)/2
- מש"ל ☺

אז מה צריך לתכנת?

- כל מה שדיברנו עליו היום ☺
- בעצם, ננסה לפתור את בעיית הבקבוקים המלאה משבוע שעבר, ובדרך נצטרך לממש את כל מה שדיברנו עליו היום

בהצלחה ☺