ברוכים הבאים לתרגול 6 ☺

שחר אנגל
shaharbel0@gmail.com
תרגול- ימי שני 14-16 וימי חמישי 13-15

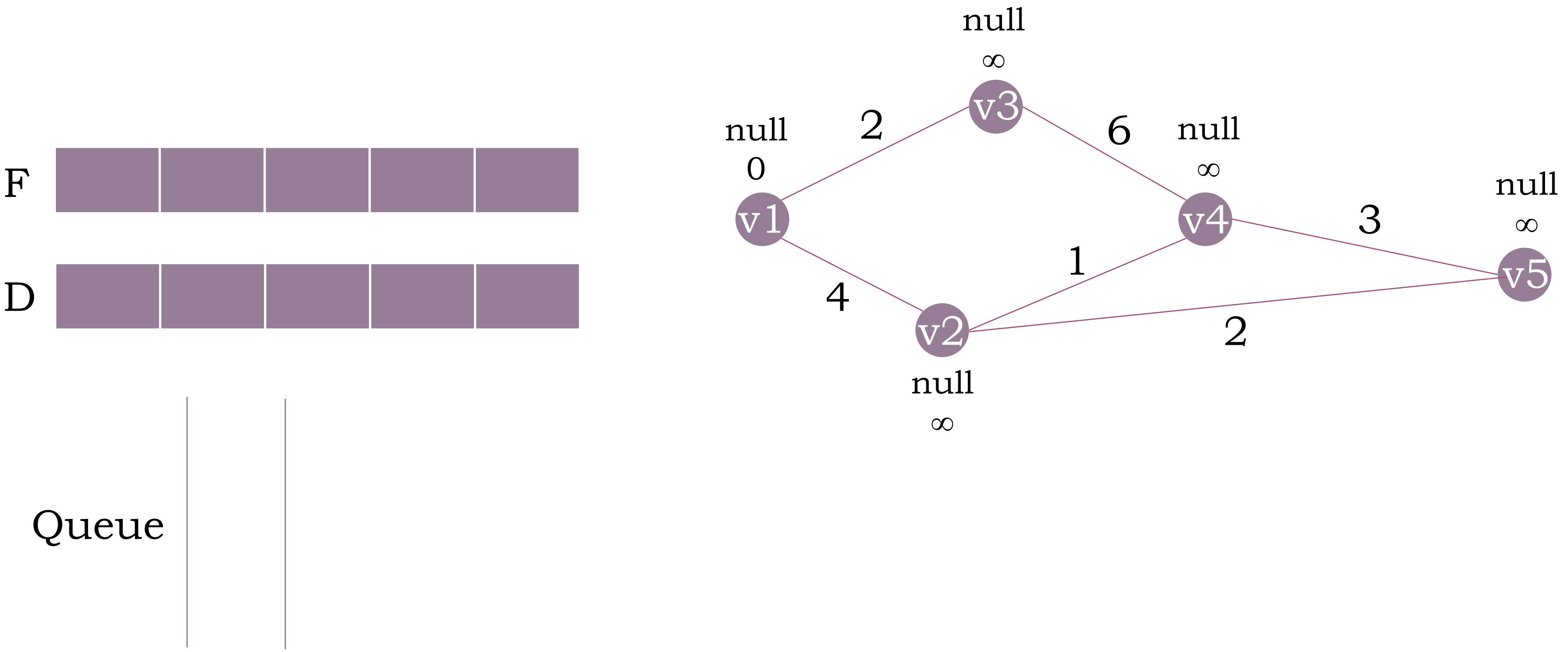
נושאי התרגול

- Dijkstra- דייקסטרה
- Bidirectional Dijkstra- דייקסטרה דו כיווני

דייקסטרה

- דייקסטרה הוא אלגוריתם חמדן. המטרה שלו היא לחשב את המסלול הקצר ביותר מקודקוד אחד לכל שאר הקודקודים. (בניגוד לFW שמחשב לנו את המסלול הקצר ביותר מכל הקודקודים לכל הקודקודים).
- איך הוא עובד?
- לכל צלע בין 2 קודקודים יש משקל
- נאתחל את הערך של קודקוד ההתחלה ל-0 ואת שאר הקודקודים ל-∞
- נתחיל לחשב את המסלול הזול ביותר באופן הבא:
- נכניס את קודקוד ההתחלה והשכנים שלו לתור. נעדכן את המרחק מקודקוד ההתחלה לכל שכן ונוציא את קודקוד ההתחלה.
- נבחר את הקודקוד המינימלי בתור, נעדכן את המרחק ממנו אל השכנים שלו, נכניס אותם לתור ונוציא אותו.
- נמשיך כך עד שנוציא את כל הקודקודים מהתור
- כעת, הערך שיש בכל קודקוד הוא המסלול הקצר ביותר מקודקוד ההתחלה אליו.

- **דייקסטרה**
- ואם נרצה לשחזר את המסלול עצמו? מה נעשה?
- נוכל לשמור עבור כל קודקוד מי האבא שלו. ניצור מערך של אבות ובכל פעם שנאתחל קודקוד מסוים בערך חדש, נרשום במערך מאיפה הגענו אליו.
- <u>לדוגמא:</u>



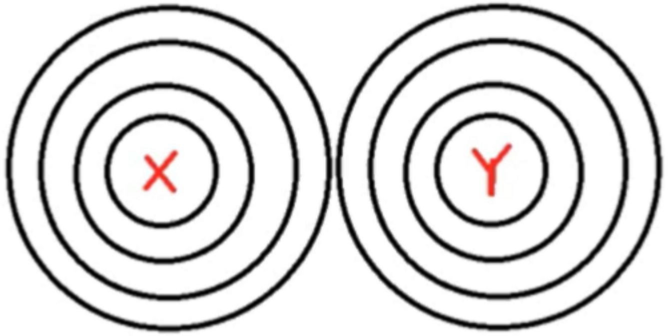
- דייקסטרה
- פסאודו קוד:

- <u>Dijkstra(G, v0):</u>
- for each v in V:
  - d[v] = ∞
  - f[v] = null
- d[0] = 0
- Queue q ← V
- while q not empty:
  - u ← q.dequeue() //שליפה של הקודקוד המינימלי
  - for each w in N(u):
    - if d[w] > d[u]+Mat[u,w]
      - d[w] = d[u]+Mat[u,w]
      - f[w] = u
- return f,d

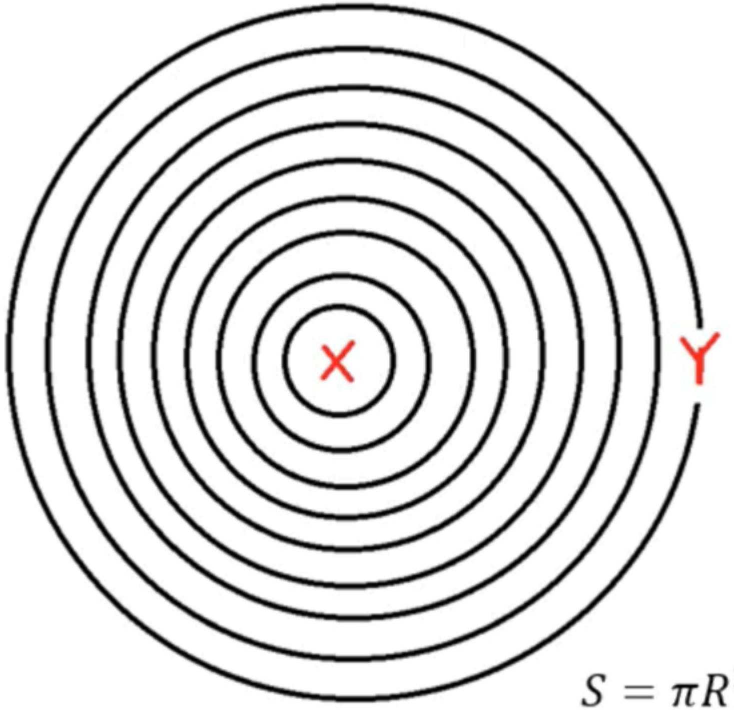
- **דייקסטרה**
- אם נרצה למצוא מסלול מקודקוד ההתחלה לקודקוד מסוים מה נעשה?
  - נשנה את תנאי עצירת האלגוריתם שיעצור ברגע שאנו מסיימים לחשב את המרחק של קודקוד זה והוא יוצא מהתור.
- נוכל לשפר את דייקסטרה בצורה הבאה:

דייקסטרה דו כיוונית

- דייקסטרה דו כיוונית היא דייקסטרה שמתקדמים בה גם מקודקוד ההתחלה וגם מקודקוד הסיום בו זמנית.
- הרעיון שעומד מאחורי זה הוא שבדייקסטרה בכל איטרציה אנו מתקדמים בשכבות של הבנים וכך נוצרים מעין מעגלים מסביב לקודקוד.
- אם נסמן את המרחק הקצר ביותר בין X ל-Y ב-R, השטח שנצטרך לבדוק הוא:



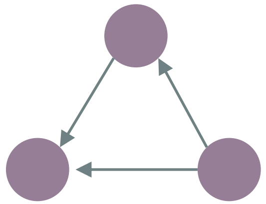
$$S = 2\left(\pi * \left(\frac{R}{2}\right)^2\right) = \frac{\pi R^2}{2}$$

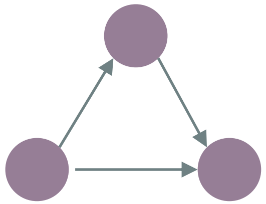


$$S = \pi R^2$$

▪ דייקסטרה דו כיוונית

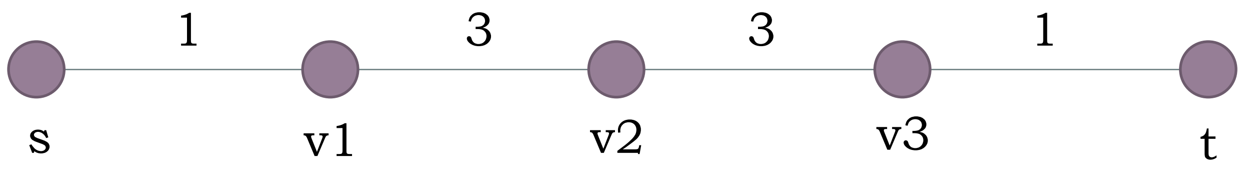
- נגדיר רגע גרף רוורס:
- <u>הגדרה:</u> גרף רוורס G^R עבור גרף G, הוא גרף עם קבוצת קודקודים V וקבוצה של הרוורס של הצלעות E^R , כך שכל צלע E ϵ(u,v) קיימת צלע E^R ϵ (v,u) ולהפך.
- <u>לדוגמא:</u>



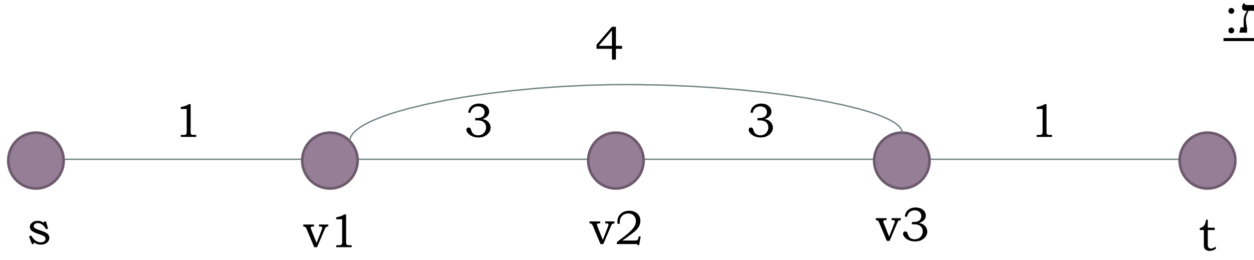


- כעת, לאחר שהבנו את ההגדרה, ננסה לחשוב איך זה עוזר לנו בפתרון דייקסטרה דו כיוונית.
- אכן, נוכל להשתמש בזה כך:
- נבנה G^R
- נריץ את האלגוריתם של דייקסטרה מ-s ב-G ומ-t ב-G^R
- נחליף לסירוגין בין שלבי דייקסטרה ב-G וב-G^R
- נעצור בשלב שקיים קודקוד v כלשהו שטופל (יצא מהתור) גם ב-G וגם ב-G^R
- לבסוף, נחשב את המסלול הקצר ביותר בין s ל-t.

- **דייקסטרה דו כיוונית**
- אבל האם בהכרח המסלול הקצר ביותר עובר דרך קודקוד v זה?
- <u>לדוגמא:</u>



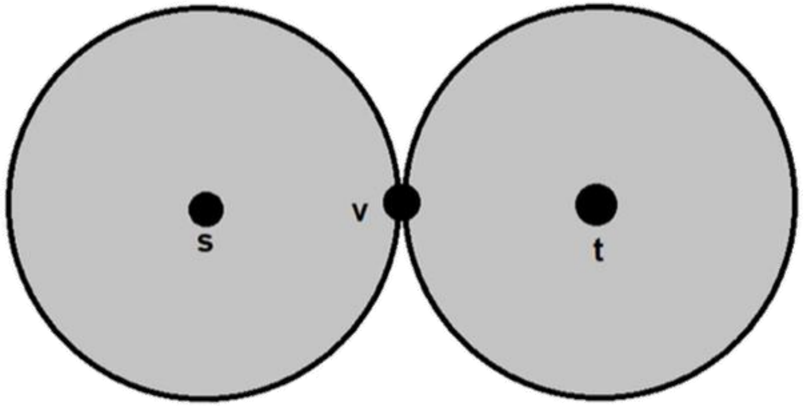
- כאן אנו עוצרים בקודקוד v2 ובהכרח עוברים במסלול הקצר ביותר
- <u>דוגמא נוספת:</u>



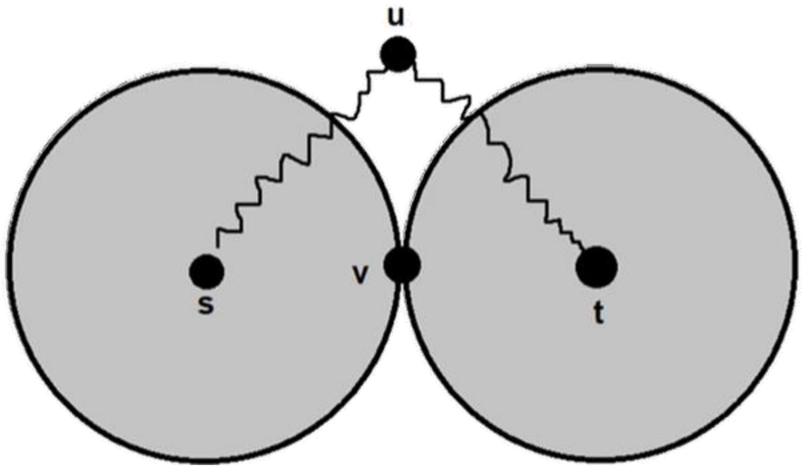
- כאן אנו גם עוצרים בקודקוד v2 אך לא מקבלים את המסלול הקצר ביותר..
- כדי לענות על השאלה נראה למה:

דייקסטרה דו כיוונית

- נסמן ב-d[u] את המרחק באלגוריתם דייקסטרה החל מקודקוד s ב-G ונסמן ב-d^R[u] את המרחק באלגוריתם דייקסטרה החל מקודקוד t ב-G^R
- <u>למה:</u> לאחר שקודקוד מסוים v טופל גם ב-G וגם ב-G^R, אזי קיים מסלול קצר ביותר מ-s ל-t שעובר דרך קודקוד כלשהו u שטופל ב-G או ב-G^R או בשניהם והמשוואה הבאה מתקיימת: $d(s,t) = d[u] + d^R[u]$
- כלומר, אחרי שמצאנו קודקוד v שטופל ב-2 האלגוריתם נוכל למצוא קודקוד u שגם טופל כבר בשני האלגוריתמים וגם נמצא על המסלול הקצר ביותר. עבור אותו קודקוד u המשוואה מתקיימת.
- נוכיח את הלמה:
- נניח שקיימים לנו 2 מעגלים הנוצרים מריצת האלגוריתם וקודקוד v הוא נקודת המפגש ביניהם.

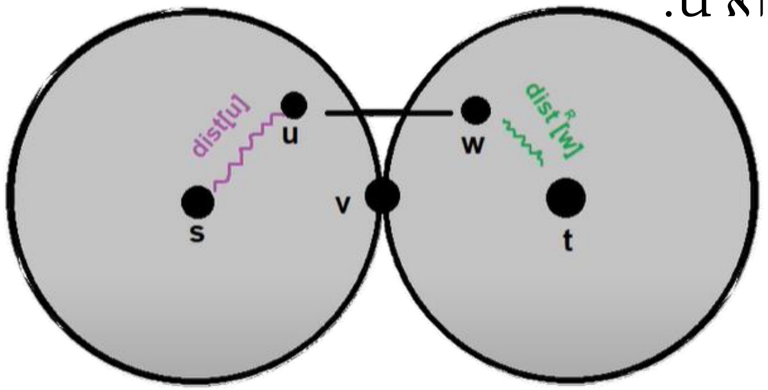


- **דייקסטרה דו כיוונית**
- ראשית נוכיח שאם קיים קודקוד u שלא טופל ע"י אף אחד מהאלגוריתמים ודרכו עובר המסלול הקצר ביותר, אז קיים לפחות קודקוד נוסף שהמסלול הקצר ביותר עובר דרכו- קודקוד v.
- אם באמת יש קודקוד u שלא טופל בכלל, אז קיים קודקוד v נוסף שטופל ע"י האלגוריתמים והמסלול הקצר ביותר עובר דרכו. בהכרח המרחק שיש ב-v נכון כי אחרי שהוא טופל הוא התקבע למסלול הזול ביותר בשני הכיוונים.



דייקסטרה דו כיוונית

- מקרה נוסף שנרצה להוכיח הוא כאשר אין קודקודים שלא טופלו בשני האלגוריתמים חוץ מנקודת המפגש בין שני המעגלים- v.
- נניח שיש מסלול זול בין s ל-t ונניח שהקודקוד האחרון שטופל באלגוריתם הישיר הוא u.



- הקודקוד הבא במסלול הקצר ביותר יהיה w כי הוא לא טופל באלגוריתם הישיר (כי u האחרון שטופל בו) ומצד שני לא יכול להיות שהוא לא טופל כלל כי הוא על המסלול הקצר ביותר. ולכן w חייב להיות בתוך המעגל של t והוא בהכרח טופל ע"י האלגוריתם ההפוך.
- לכן, המרחק כרגע בין s ל-t הוא: $d(s,t) = d[u] + l(u,w) + d^R[w]$. נרצה להוכיח שזה שווה למשוואה מהלמה:
- $d(s,t) = d[u] + d^R[u]$
- בעצם אנו מוכיחים שנקודת המפגש v לא בהכרח נמצאת על המסלול הזול ביותר אלא שיהיה קודקוד אחר u שהמרחק מ-s אליו והמרחק ממנו ל-t יהיה הקצר ביותר.

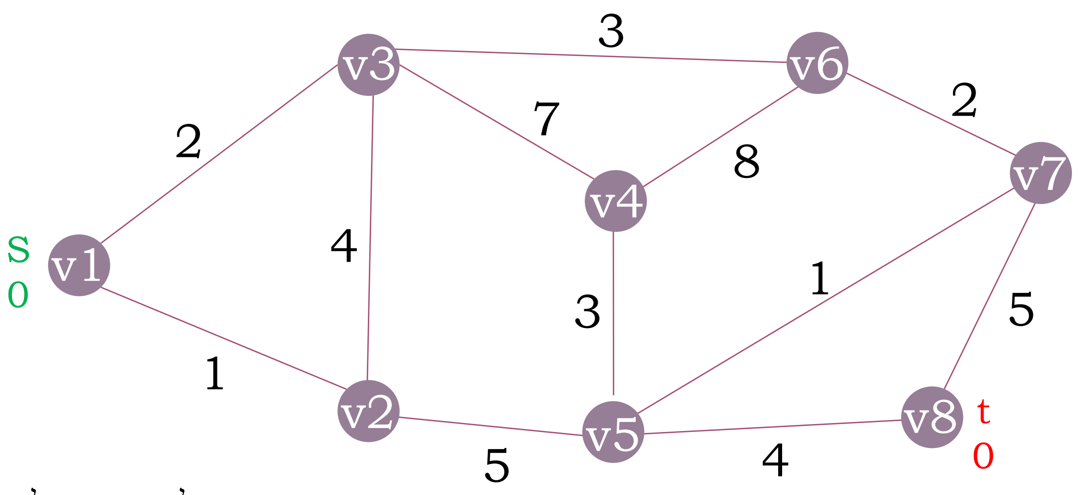
דייקסטרה דו כיוונית

- $d(s,t) = d[u] + l(u,w) + d^R[w]$
- $d(s,t) = d[u] + d^R[u]$
- בשתי המשוואות יש $d[u]$ ולכן כדי להשוות אותן נצטרך להשוות בין $d^R[w] + l(u,w)$ לבין $d^R[u]$
- נשים לב ש- $d(u,t) \le d^R[u]$ כי בכל רגע של הפעלת האלגוריתם המרחקים רק משתפרים.
- מצד שני, w טופל ע"י האלגוריתם ההפוך ולכן הוא 'הקל' על הצלע (u,w) שיוצאת ממנו ולכן $l(u,w) + d^R[w] \ge d^R[u]$
- בעצם, $l(u,w) + d^R[w]$ הוא המרחק הקצר ביותר בין u ל-t שעובר דרך w, ולכן לפי כלל הסנדוויץ' $d^R[u]$ שווה למרחק הקצר ביותר בין u ל-t.
- משל ☺

דייקסטרה דו כיוונית

- כעת נממש:
- נבצע החלפות בין האלגוריתמים בכל איטרציה עד שנפגוש קודקוד שטופל בשניהם.
- ניקח את כל הקודקודים שטופלו ונחפש מינימום בסכום של האלגוריתם הישיר וההפוך. אותו מספר מינימלי יהיה המרחק הזול ביותר.
- כדי לייצר את המסלול עצמו נשרשר את המסלול מ-s לאותו קודקוד מינימלי והרוורס של המסלול מ-t לאותו קודקוד.

- <u>דוגמת הרצה:</u>

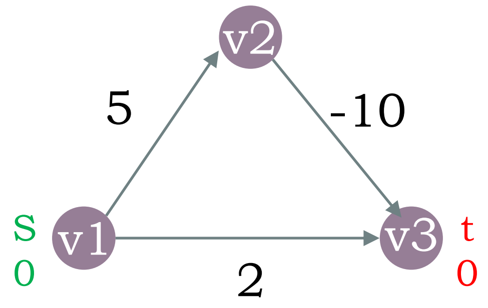


כלומר, העלות הזולה ביותר היא 10
והמסלול הקצר ביותר הוא v1-v2-v5-v8

- **דייקסטרה דו כיוונית**
- לסיכום, הסיבוכיות של דייקסטרה דו כיוונית היא בדיוק כמו הסיבוכיות של דייקסטרה רגילה.
- מבחינת זמן ריצה- זה יכול להיות פי 2 יותר מהיר
- מבחינת זיכרון- תופסים פי 2 אחסון כי צריך לייצר גם את הגרף ההפוך.

משקלים שליליים

- למדנו על אלגוריתם FW שמוצא מסלולים קצרים ביותר מכל קודקוד לכל קודקוד. הסברנו שהוא יכול להתמודד עם משקלים שליליים אבל לא עם מעגלים שליליים. (ברגע שיש מעגל שלילי ברף השאלה של מה המסלול הקצר ביותר כבר לא רלוונטית)
- האם דייקסטרה יכול להתמודד עם משקלים שליליים?
  - לא
- <u>לדוגמא:</u>



- אנו רואים שהוא לא מצא את המסלול הקצר ביותר: 5-, לעומת FW שכן היה מוצא..

אז מה צריך לתכנת?

- כל מה שדיברנו עליו היום ☺

1. דייקסטרה
2. דייקסטרה דו כיווני
3. בונוס: כשלמדנו את האלגוריתם של FW שאלנו איך הוא יכול להתמודד עם משקלים על קודקודים/קודקודים וצלעות. נשאל את אותה השאלה על דייקסטרה: איך הוא יתמודד אם המשקלים יהיו על הקודקודים? ועל הקודקודים והצלעות?
מוזמנים לחשוב, לממש, ולהוכיח.

בהצלחה ☺