**תרגול OOP – 8.11**

**תכנון דינמי:**

* **הרעיון** הוא שיש לנו בעיה שפתרונה רקורסיבי, אבל במקום לקרוא לרקורסיה שוב ושוב אנו מפרקים את הבעיה לתתי בעיות ושומרים בצד את חישובי הביניים שאנו מבצעים
* יכול להוריד זמן ריצה מאקספוננציאלי לפולינומיאלי
* ראשית נוודא שיש פתרון רקורסיבי וניתן לפרק לתתי בעיות נבצע:
  + כמה צעדים יש לנו – יגדיר את גודל הטבלה שלנו לשמירת חישובי הביניים
  + איך ממלאים את הטבלה כך שבסוף נקבל את הפתרון שנרצה
  + איך נחלץ את הפתרון
* דוגמא – פיבונצ'י
  + נשמור בטבלה את הערכים הראשוניים ואז במקום לקרוא רקורסיבית נמלא את המערך לפי הסדר בעזרת התאים הקודמים שכבר מלאים
* **תרגיל** – נתון מערך MxN כשבכל תא כתוב לנו את העלות של המעבר באותו תא.

נרצה למצוא את המסלול עם העלות הנמוכה ביותר מהפינה השמאלית העליונה לפינה הימנית התחתונה:

* + פתרון נאיבי – לעבור על כל האפשרויות למסלולים – יקר מאוד
  + עיקרון האופטימליות של בלמן – אם נרצה שמשהו יהיה אופטימלי אז כל חלק שלו גם צריך להיות אופטימלי.
  + פתרון דינמי –
    - נגדיר טבלה בגודל MxN ונאתחלה אותה באפסים
    - כל תא יכיל את מחיר המסלול המינימלי מהפינה השמאלית העליונה לתא הנוכחי. כך בפינה הימנית התחתונה יהיה לנו את המחיר המינימלי של המסלול הרצוי
    - אם אנחנו בעמודה השמאלית או בשורה העליונה אז בהכרח הגענו מהתא הקודם לנו בציר זה ולכן נשים את הערך שבתא הקודם ועוד ערך התא הנוכחי.
    - במקרה הכללי – נשים את הערך המינימלי של להגיע מהתא השמאלי או מהתא העליון.
* **בעיית התרמיל השברי –**
  + מעבר על כל האפשרויות - בעיה NP קשה (לוקח המון זמן לפתור אותה). לכן נוסיף את ההקלות:
    - W פולינומי ב-n
    - משקל המוצרים וקיבולת התרמיל הם מספרים טבעיים.

עכשיו ניתן לפתור את הבעיה בתכנון דינאמי.

* + פתרון דינמי – נניח שיש לנו סדר על המוצרים.
    - נגדיר מערך בגודל n x (W+1)
    - בתא הi,j יהיה לנו את השווי המקסימלי של סחורה שיכולנו לגנוב עד הסחורה ה-i אם הגבל המשקל היא j בזמן שהגענו למוצר ה-i
    - מילוי הטבלה –
      * אם משקל המוצר הראשון גדול מהקיבולת הנוכחית נכניס 0, אם הוא קשטן שווה נכניס את ערך המוצר הראשון
      * אם אנו במוצר i – אם המשקל של המוצר הנוכחי גדול מהקיבולת נמלא את הטבלה עם ערך המוצר הקודם (לא עדכנו את שווי התיק), אבל אם המשקל קטן מהקיבולת ניקח את המקסימלי בין להכניס את המוצר (בהתייחסות לתא עם קיבולת קטנה יותר במשקל הנוכחי, ונלך אחורה במוצרים)או לא להכניס.

**ממשקים:**

* ממשק הוא חוזה שמחלקות שמממשות את הממשק צריכות לעמוד בו – לא משנה איך ימומשו הפונקציות אבל הן חייבות להיות ממומשות.
* התנהגות דומה למחלקות שונות
* מאפשר פולימורפיזם כי אפשר להסתכל על דברים שונים מאותם משקפיים
* מחלקה יכולה לממש כמה ממשקים שונים
* ממשק יכול להרחיב ממשקים אחרים – נוסיף בשורת ההצהרה extands inter1, inter2,..
  + מאפשר להתייחס למוצרים שממשים את אותם ממשקים בתור יחידה אחת
  + נכתוב שהמחלקה מממשת את הממשק המרחיב וזה יגיד שהיא מממשת את כל הממשקים שהוא מרחיב.

**Enum:**

* כמו מחלקה, יכול לממש ממשקים.
* בתוך מחלקה הENUM ניתן לממש מתודות
* בתחילת הENUM נכתוב את איבריו (שמותיהם באותיות גדולות)
* אם יש לנו בנאי, אחרי כל שם של איבר בENUM נכתוב סוגריים ואת הפרמטרים שנרצה לשלוח לבנאי ביצירת האיבר הנוכחי של ה-ENUM.
  + למעשה יהיו לנו אובייקטים מוגדרים מראש של איברי הENUM כמו בENUM רגיל – הקונסטרקטור לא ייקרא כל פעם מחדש אלא ביצירת הENUM הוא יקרא 3 פעמים ויישמרו האובייקטים באיברי הENUM
* ניגש לאיברי הENUM באותה צורה ובנוסף נוכל לגשת על כל אחד מהם לפונקציות