

אלגוריתמים 1

המרצים: ספי נאור ויובל רבני
שעות קבלה: אחרי השעור או בתאום מראש

קביעת הציון: 90% ציון הבחינה + 10% תרגילי בית

ספרות:

- J. Kleinberg and É. Tardos, Algorithm Design, Pearson Education, 2006.
- S. Dasgupta, C.H. Papadimitriou, and U. Vazirani, Algorithms, McGraw-Hill, 2006.
- S. Even, Graph Algorithms, Computer Science Press, 1979.
- T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms (2nd edition), The MIT Press & McGraw-Hill, 2001.

מהו אלגוריתם?

”הגדרה“: אלגוריתם הוא מתכון לפתרון של בעייה מתמטית במספר סופי של צעדים בסיסיים ומוגדרים היטב.

דוגמה: חיבור או כפל שני מספרים בייצוג עשרוני כפי שנלמד בבי”ס יסודי.

נתונה פונקציה $f: I \rightarrow O$.
בהינתן $x \in I$ נרצה לחשב את $f(x)$.
הקלט לאלגוריתם הוא x , הפלט הוא $f(x)$.

בעיות מעניינות רבות אינן מוגדרות היטב מתמטית.
למשל: זיהוי תמונות, הבנת שפה.

אלגוריתמים הם מושג עתיק:
הבבלים תיעדו אלגוריתמים לחישובים שונים.
אלגוריתם אוקלידס למציאת ממג"ב.
אריתמטיקה עשרונית.

הגדרה מתמטית: שנות ה-30.
יש מספר הגדרות שקולות (מודלים חישוביים).

הנחה: | ו-O הן קבוצות דיסקרטיות, בפרט הקלט ניתן
להצגה במספר סופי של ביטים.



- זיכרון אינסופי של רגיסטרים: כתובת של רגיסטר היא מספר שלם.
- כל רגיסטר מחזיק מספר שלם.
- הקלט הוא סדרת מספרים על סרט קלט לקריאה בלבד (המספר הראשון: אורך הקלט).
- הפלט נרשם על סרט פלט לכתיבה בלבד.
- התוכנית היא סדרה קבועה של פקודות המבוצעות לפי הסדר.

הפקודות המותרות

$r[i] = 0$

$r[i]++$

$r[i]--$

$r[i] = r[j]$

$r[i] += r[j]$

$r[i] -= r[j]$

$r[i] = r[r[j]]$

if ($r[i] \leq 0$) goto p

read($r[i]$)

write($r[i]$)

goto p

$r[i] *= r[j]$

$r[i] /= r[j]$

דוגמה - חישוב מקסימום

```
1. read(r[0])
2. read(r[1])
3. r[0]--
4. if (r[0] <= 0) goto 11
5. read(r[2])
6. r[3] = r[2]
7. r[3] -= r[1]
8. if (r[3] <= 0) goto 3
9. r[1] = r[2]
10. goto 3
11. write(r[1])
```

רגיסטר $r[0]$ ימנה את הקלט
רגיסטר $r[1]$ יחזיק את התוצאה

רגיסטר $r[2]$ יחזיק את הקלט הנוכחי

רגיסטר $r[3]$ ישמש לבדיקה אם
הקלט הנוכחי גדול מהמקסימום
עד כה

- קשה לתאר אלגוריתם במודל RAM, לכן נשתמש בשפה עשירה יותר:

- משתנים, מבני נתונים, ביטויים מורכבים
- תנאים בוליאניים, לולאות, פרוצדורות

- לרוב נדלג על קריאת הקלט וכתיבת הפלט; נסתפק בתאור הביצוע של עיקר החישוב.

- התרגום לתוכנית RAM צריך להיות ברור.

```
read(n)
read(max)
for i ← 2 to n do
  read(next)
  if (next > max) max ← next
end for
write(max)
```

זהו מודל מופשט של (תוכנית) מחשב.
מעבד אחד, הקלט כולו זמין בהתחלה.

מודלים חלופיים: חישוב מקבילי/מבוזר, מערכות
ריאקטיביות, זיכרון חיצוני, ...

לא נדון במודלים הללו בקורס זה.

ניתוח אלגוריתמים

- נכונות
- ניתוח משאבי החישוב הדרושים לפתרון:
 - זמן
 - מקום (בנוסף לקלט)
- 👉 נתמקד בנייתו אסימפטוטי של משאבי החישוב כפונקציה של אורך הקלט.

הביצועים תלויים, כמובן בייצוג הקלט והפלט.
למשל: קובץ zipped אולי קצר יותר מהמקור.

סיבות להתעניין בניתוח אסימפטוטי

- מאפיין את התנהגות האלגוריתם על קלט ארוך מאוד. (לדוגמה: DNA אנושי כולל כ- 10^9 נוקלאוטידים. אינדקס של מנוע חיפוש כולל כ- 10^{10} מסמכי HTML.)
- לא תלוי בחומרה ספציפית.

המודל הבסיסי

זמן: סך # הפעולות שביצענו.
מקום: סך # הרגיסטרים שהשתמשנו בהם.

חיוב לפי ביטים

(מדמה מגבלות נוספות של מחשב מציאותי)

גישה לרגיסטר עולה # הביטים בכתובת הרגיסטר + #
הביטים בתוכן הרגיסטר.
אורך הקלט נמדד לפי מספר הביטים.