האוניברסיטה הפתוחה

מבני נתונים ומבוא לאלגוריתמים
מפגש הנחיה מס' 2

מדעי המחשב, קורס מס' 20407

סמסטר 2016ב

מנחה: ג'ון מרברג

מה ראינו במפגש הקודם?

- מושגים בסיסיים כגון: בעיה, אלגוריתם, יעילות ועוד.
- טכניקות לבניית אלג': לולאות, הפרד ומשול, רקורסיה
- הוכחת נכונות של אלג': שמורת לולאה, אינדוקציה
- ניתוח זמנים: ספירת פעולות, סדר גודל, מקרה גרוע
- בעיית החיפוש: לינארי, בינארי
- בעיית המיון:
  - מיון אינקרמנטלי בזמן $O(n^2)$ – כגון מיון הכנסה
  - מיון בשיטת הפרד ומשול בזמן $O(n\log n)$ – כגון מיון מיזוג

מפגש שני

- נושאי השיעור
  - פרק 2 בספר – תרגול נוסף של סוגי אלגוריתמים
  - פרק 3 בספר – גידול של פונקציות

- תוכן העניינים
  - נתוח דוגמאות של אלגוריתם רקורסיבי ואלגוריתם איטרטיבי
  - המוטיבציה לשימוש בסימונים אסימפטוטיים
  - הגדרת הסימונים
  - חסמים שימושיים ותכונות שימושיות
  - חישוב סיבוכיות הזמן של אלגוריתם
  - תרגיל – אלגוריתמים לחישוב טור גאומטרי

מבוסס על מצגת של ברוך חייקין ואיציק בייז

מגדלי האנוי - אלגוריתם רקורסיבי

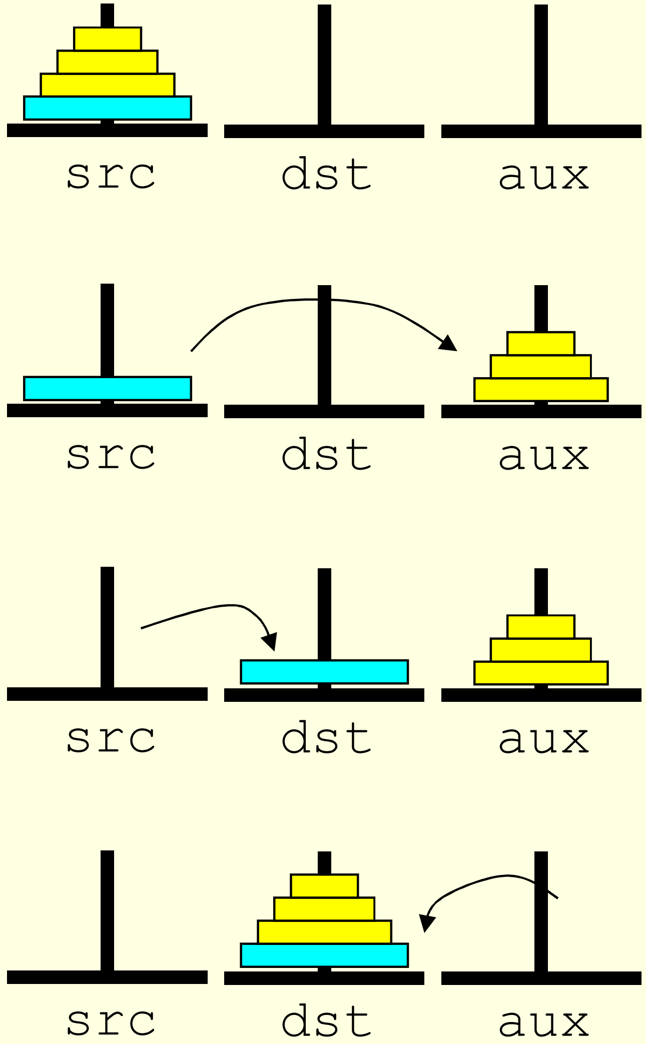
- הבעיה: העברת *n* טבעות מעמוד המקור לעמוד היעד

Hanoi(*n*, *src*, *dst*, *aux*)

Input: A source peg *src*, initially with disks 1, 2, ..., *n*; a destination peg *dst*, and an auxiliary peg *aux*.

Output: A sequence of legal moves that brings all disks to peg *dst* (legal moves – one disk at a time, and a disk can not be put on a smaller disk).

1. **if** $n > 0$
2. **then** Hannoi($n - 1$, *src*, *aux*, *dst*)
3. move disk *n* from peg *src* to peg *dst*
4. Hannoi($n - 1$, *aux*, *dst*, *src*)



כדורים בכד - אלגוריתם איטרטיבי

בכד יש m כדורים שחורים ו- n כדורים לבנים, כאשר $m>0$, $n=2k>0$ (כלומר n זוגי).
כמו כן, מחוץ לכד יש מספר בלתי מוגבל של כדורים שחורים.

בצע את הצעדים הבאים:

1. אם בכד יש פחות משני כדורים, עצור.
2. הוצא מהכד (בלי להסתכל בתוך הכד) שני כדורים כלשהם.
3. אם שני הכדורים שהוצאת הם מאותו צבע, הכנס כדור שחור אחד לכד
4. אחרת, החזר לכד את הכדור הלבן שהוצאת
5. חזור לצעד 1

<u>טענה</u>: הלולאה בשורות 1-5 עוצרת. בעצירה יש בדיוק כדור אחד בכד, וצבעו שחור.
<u>הוכחה</u>: בעזרת שמורת לולאה

<u>תרגיל</u>: הוכח כי בתחילת האלגוריתם נחוצים בדיוק k כדורים שחורים מחוץ לכד
(כלומר k הוא מספר הכרחי ומספיק).

סימונים אסימפטוטים - מוטיבציה

- זמן הריצה (מספר הפעולות) של אלגוריתמים מתואר בעזרת פונקציות מהטבעיים לממשיים החיוביים: $f: N \rightarrow R^+$
  - $f(n) = 4n^3 + n^2 \cdot \lg^5 n + 100n + 17$
  - $g(n) = n \sum_{k=1}^{n} 1/k$

  כאשר n הוא גודל הקלט

- נרצה דרך פשוטה להשוות את הגודל של פונקציות שונות
- הסימון האסימפטוטי הוא דרך פשוטה להשוות פונקציות כאלה
  - $f(n) = \Theta(n^3)$
  - $g(n) = \Theta(n \lg n)$ (כי $\sum_{k=1}^{n} 1/k$ זה טור הרמוני)

מוטיבציה - המשך

- מה שחשוב הוא **קצב הגידול** של הפונקציה – איך משתנה הפונקציה ככל ש-*n* גדל.
- מטרתנו היא לבצע "חלוקה גסה למגירות של סדרי גודל"
- לכן אפשר להתעלם מקבועים ומחזקות נמוכות.
- מתעניינים בעצם בהתנהגות האסימפטוטית של הפונקציה – איך מתנהגת הפונקציה כש-*n* שואף לאינסוף.
- כלומר, אנו מתעניינים רק בקלט הגדול או שווה ל- n_0 כלשהו.
- ניתן לחשוב על כל קלט הקטן מ- n_0 כמקרה קצה.
- כל מגירה של סדר גודל היא מחלקת שקילות של הפונקציות מאותו סדר גודל

הסימונים האסימפטוטים

- באופן **אינטואיטיבי**, הסימונים מאפשרים להשוות בין **סדרי הגודל** שתי פונק' ולומר מי "גדולה", מי "קטנה", או האם הן "שוות" (בכל היחסים מדובר בסדר גודל) :

חסם עליון	O	$\leq$
חסם תחתון	Ω	$\geq$
חסם הדוק	Θ	$=$
חסם עליון לא הדוק	o	$<$
חסם תחתון לא הדוק	ω	$>$

דוגמאות :

- $f(n)=O(g(n)), \quad h(n)=\Theta(g(n)), \quad f(n)=\omega(q(n))$
- כלומר : f קטנה שווה בסדר גודל מ- g ,
  h ו- g הן מאותו סדר גודל,
  f גדולה ממש בסדר גודל מ- q

הסימון O - חסם עליון

- הגדרה
  - $f(n) = O(g(n))$ אם *קיימים* קבועים חיוביים c ו- n_0 כך שמתקיים $f(n) \le cg(n)$ לכל $n \ge n_0$
  - משמעות: החל מנקודה n_0 הערכים של f *אינם גדולים* מאלו של g (עד כדי כפל בקבוע c)
  - אינטואיטיבית: $f \le g$ (או: g היא חסם עליון של f) כש-n שואף לאינסוף
  - כדי להוכיח ש- $f(n) = O(g(n))$, יש למצוא c ו- n_0 מתאימים

- דוגמה
  - $f(n) = 10n^2 + 30 = O(n^2)$
  - $f(n) = 10n^2 + 30 = O(n^3)$

הסימון Ω - חסם תחתון

- הגדרה
  - $f(n) = \Omega(g(n))$ אם *קיימים* קבועים חיוביים c ו- n_0 כך שמתקיים $cg(n) \le f(n)$ לכל $n \ge n_0$
  - משמעות: החל מנקודה n_0 הערכים של f *אינם קטנים* מאלו של g (עד כדי כפל בקבוע c)
  - אינטואיטיבית: $f \ge g$ (או: g היא חסם תחתון של f) כש-n שואף לאינסוף
  - כדי להוכיח ש- $f(n) = \Omega(g(n))$, יש למצוא c ו- n_0 מתאימים
- דוגמה
  - $f(n) = n^3 - 5n^2 + 10 = \Omega(n^3)$
  - $f(n) = n^3 - 5n^2 + 10 = \Omega(n^2)$

הסימון Θ (טתא) - חסם הדוק

- הגדרה
  - $f(n) = \Theta(g(n))$ אם *קיימים* קבועים חיוביים c_1, c_2 ו- n_0 כך שמתקיים $c_1 g(n) \le f(n) \le c_2 g(n)$ לכל $n \ge n_0$
  - משמעות: החל מנקודה n_0 הערכים של f קרובים לאלו של g (עד כדי כפל בקבוע)
  - אינטואיטיבית: $f = g$ (או: g בסדר גודל של f) כש-n שואף לאינסוף
  - כדי להוכיח ש- $f(n) = \Theta(g(n))$, יש למצוא c_1, c_2 ו- n_0 מתאימים

- דוגמה
  - $f(n) = 10n^2 + 100n + 30 = \Theta(n^2)$

האוניברסיטה הפתוחה

החסם o - חסם עליון לא הדוק

- הגדרה
  - $f(n) = o(g(n))$ אם <u>*לכל קבוע*</u> חיובי c *קיים* קבוע חיובי n_c כך שמתקיים $f(n) < cg(n)$ לכל $n \geq n_c$
  - משמעות: החל מנקודה n_c *הערכים* של f <u>*קטנים ממש*</u> מאלו של g (עד כדי כפל בקבוע c)
  - אינטואיטיבית: $f < g$ (או: g היא חסם עליון <u>לא הדוק</u> של f) כש-n שואף לאינסוף
  - כדי להוכיח ש- $f(n) = o(g(n))$, יש למצוא n_c מתאים כפונקציה של c
- הגדרה שקולה
  - $f(n) = o(g(n))$ אם $\lim_{n\to\infty} f(n)/g(n) = 0$
- דוגמה:
  - $f(n) = 2n^2 = o(n^3)$

החסם ω - חסם תחתון לא הדוק

- הגדרה
  - $f(n) = \omega(g(n))$ אם לכל קבוע חיובי c *קיים* קבוע חיובי n_c כך שמתקיים $cg(n) < f(n)$ לכל $n \geq n_c$
  - משמעות: החל מנקודה n_c *הערכים* של f *גדולים ממש* מאלו של g (עד כדי כפל בקבוע c)
  - אינטואיטיבית: $f > g$ (או: g היא חסם תחתון לא הדוק של f) כש-n שואף לאינסוף
  - כדי להוכיח ש- $f(n) = \omega(g(n))$, יש למצוא n_c מתאים כפונקציה של c
- הגדרה שקולה
  - $f(n) = \omega(g(n))$ אם $\lim_{n\to\infty} f(n)/g(n) = \infty$
- דוגמה:
  - $f(n) = n^2 = \omega(\text{nlgn})$

חסמים הדוקים

- חסמים הדוקים
  - אם $f(n) = O(g(n))$ וגם $f(n) \neq o(g(n))$ אז $f(n) = \Theta(g(n))$
  - אם $f(n) = \Omega(g(n))$ וגם $f(n) \neq \omega(g(n))$ אז $f(n) = \Theta(g(n))$
  - $f(n) = \Theta(g(n))$ אם"ם $f(n) = O(g(n))$ וגם $f(n) = \Omega(g(n))$

חסמים שימושיים

- טור חשבוני

  $A(n) = \Sigma_{k=1}^{n} k = ½n(n + 1) = \Theta(n^2)$
- טור גאומטרי מתכנס ($|x| < 1$)

  $G(n, x) = \Sigma_{k=0}^{n} x^k = (x^{n+1} – 1) / (x – 1) = \Theta(1)$
- פולינום מדרגה d ($a_d \neq 0$)

  $P_d(n) = \Sigma_{k=0}^{d} a_k n^k = \Theta(n^d)$
- חזקות של לוגריתמים

  $\log^k n = o(n)$ לכל $k > 0$
- הטור ההרמוני

  $H_n = \Sigma_{k=1}^{n} 1/k = \Theta(\log n)$
- מספרי פיבונאצ'י ($F_n = F_{n-1} + F_{n-2}$, $F_0 = 0$, $F_1 = 1$)

  $F_n = \Theta(\varphi^n)$, $\varphi = ½(\sqrt{5} + 1) \cong 1.618$
- עצרת

  $n! = o(n^n)$

  $\log(n!) = \Theta(n\log n)$

תכונות שימושיות

- טרנזיטיביות החסם העליון

  אם $f(n) = O(g(n))$ וגם $g(n) = O(h(n))$

  אז $f(n) = O(h(n))$

  (באופן דומה קיימת טרנזיטיביות של Ω ושל Θ)

- חסם עליון על סכום פונקציות

  אם $f_1(n) = O(g_1(n))$ וגם $f_2(n) = O(g_2(n))$

  אז $f_1(n) + f_2(n) = O(\max\{g_1(n), g_2(n)\})$

- חסם עליון על מכפלת פונקציות

  אם $f_1(n) = O(g_1(n))$ וגם $f_2(n) = O(g_2(n))$

  אז $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$

יש תכונות נוספות בספר ובחוברת הלמידה

האוניברסיטה הפתוחה

חישוב סיבוכיות הזמן של אלגוריתם

- פעולות פשוטות
  - ביצוע פעולות חשבון אלמנטריות: $\Theta(1)$
  - השמה והשוואה: $\Theta(1)$
  - יצירת אובייקט בגודל כלשהו: $\Theta(1)$

- פעולות מורכבות
  - קריאה לפונקציה שעלותה חסומה ע"י $O(g(n))$, עם קלט בגודל m: $O(g(m))$
  - סדרה של k פעולות (k קבוע) שעלות כל אחת מהן i חסומה ע"י $O(g_i(n))$, עם קלט בגודל m: $O(\max\{ g_i(m) \})$
  - לולאה בת k חזרות על פעולה שעלותה חסומה ע"י $O(g(n))$, עם קלטים בגודל m_i: $\sum_{i=1}^{k} O(g(m_i))$

תרגיל: חישוב טור גאומטרי

- נתחו את זמן הריצה של שלושת האלגוריתמים הבאים לחישוב הטור הגאומטרי $\Sigma_{i=0}^{n} x^i = 1 + x + \ldots + x^n$
- פתרון א' – חשב וסכום את $n + 1$ האיברים מסוג x^i

```
GeometricSeriesSum1(x, n)
sum ← 0
for i ← 0 to n
    do prod ← 1
       for j ← 1 to i
           do prod ← prod * x
       sum ← sum + prod
return sum
```

<u>Initialization and Finalization</u>
$\Theta(1)$ steps

<u>Outer loop</u>
add the ith term to the sum, $i = 0, \ldots, n$

<u>Inner loop</u>
compute the term x^i in $\Theta(i)$ steps

Summary
$T(n) = \Theta(1) + \Sigma_{i=0}^{n}\Theta(i) = \Theta(n^2)$

תרגיל: טור גאומטרי (המשך)

- פתרון ב' – השתמש בכלל הורנר $\Sigma_{i=0}^{n} x^i = (\Sigma_{i=0}^{n-1} x^i)x + 1$

```
GeometricSeriesSum2(x, n)
sum ← 0
for i ← 0 to n
    do sum ← sum * x + 1
return sum
```

<u>Initialization and Finalization</u>
$\Theta(1)$ steps

<u>Loop</u> (for $i = 0, .. n$)
multiply by x and increment by 1

Summary
$T(n) = \Theta(1) + \Sigma_{i=0}^{n}\Theta(1) = \Theta(n)$

האוניברסיטה הפתוחה

תרגיל: טור גאומטרי (המשך)

- פתרון ג' – בצע חישוב ישיר $\Sigma_{i=0}^{n} x^i = (x^{n+1} - 1) / (x - 1)$
- חישוב חזקה x^n (n שלם): $x^{2m+1} = x \cdot (x^2)^m$, $x^{2m} = (x^2)^m$, $x^0 = 1$, $0^n = 0$

GeometricSeriesSum3(x, n)
1. **if** $x = 1$
2. **then return** $n + 1$
3. **else return** (Power(x, $n + 1$) – 1) / (x – 1)

Summary

$T(n) = \Theta(1) + P(n + 1)$

$= \Theta(\lg n)$

Power(x, n)
1. **if** $x = 0$ **then return** 0
2. **else if** $n = 0$ **then return** 1
3. **else if** n is even
4. **then return** Power(x * x, n / 2)
5. **else return** x * Power(x * x, (n – 1) / 2)

Power

$P(n) = P(n / 2) + \Theta(1)$

$= \ldots$

$= \Theta(\lg n)$