ID: 207884883
Ron Nuriel

Question 1 (15 points)

Prove the following claim using **induction**:

Claim: For all $n \geq 1$ the number $7^n + 3^n - 2$ is divisible by 8.

נסמן $P(n) = 7^n + 3^n - 2$

בסיס האינדוקציה

מתקיים.
$n=1$ $\quad 7+3-2 = 8 \Rightarrow$ מתחלק ב 8.
$n=2$ $\quad 7^2+3^2-2 = 56 \Rightarrow$ מתחלק ב 8

צעד האינדוקציה $(n \to n+2)$

נניח כי $P(n)$ נכון, כלומר $P(n) \bmod 8 \equiv 0$ ונראה כי $P(n+2)$ נכון:

$$7^{n+2} + 3^{n+2} - 2 = 49 \cdot 7^n + 9 \cdot 3^n - 2 \equiv$$
$$(49 \bmod 8) \cdot 7^n + (9 \bmod 8) 3^n - 2 \pmod 8 \equiv$$
$$1 \cdot 7^n + 1 \cdot 3^n - 2 \equiv 0 \bmod 8$$

כלומר אם $P(n)$ נכון אז $P(n+2)$ נכון.

הטענה נכונה לשני ערכי בסיס עוקבים $n=1,2$
ומתוך כל n עוברת ל $n+2$ נקבל באינדוקציה כי $P(n)$
נכונה לכל $n \geq 1$.

Question 2 (15 points)

Let $P(n) = \frac{1}{3}n^5 - 10n^3 + 8n - 24$. Formally prove that $P(n) = \Theta(n^5)$.

Do not use the claim about polynomial that was shown in class.

נרצה להוכיח כי $P(n) = O(n^5)$ וגם כי $P(n) = \Omega(n^5)$

חסם עליון: לכל $n \geq 1$

$$\frac{1}{3}n^5 - 10n^3 + 8n - 24 \leq \frac{1}{3}n^5 + 10n^3 + 8n + 24$$

$$\frac{1}{3}n^5 - 10n^3 + 8 - 24 \leq \frac{1}{3}n^5 + 10n^5 + 8n^5 + 24n^5 \leq \frac{127}{3}n^5$$

$$\boxed{P(n) \geq \frac{127}{3}n^5}$$

$$\boxed{c = 43 \quad n_0 = 1}$$

חסם תחתון:

$$\frac{1}{3}n^5 - 10n^3 + 8n - 24 \geq \frac{1}{3}n^5 - 10n^3 - 24n^3 = \frac{1}{3}n^5 - 34n^3$$

$$\frac{1}{3}n^5 - 10n^3 + 8n - 24 \geq n^3\left(\frac{1}{3}n^2 - 34\right)$$

נבחר n גבול בשביל שהביטוי בסוגריים יהיה גדול

$$\frac{1}{3}n^2 - 34 \geq \frac{1}{6}n^2$$

$$2n^2 - 204 \geq n^2$$

$$n^2 \geq 204$$

$\boxed{n_1 = 14.2} \Rightarrow$ ולכן נבחר $n_0 = 15$

$n_2 = \cancel{-14.2}$ $\quad n \geq 1$

$$P(n) \geq n^3 \cdot \left(\frac{1}{6}n^2\right) = \frac{1}{6}n^5$$

$$P(n) \geq \frac{1}{6}n^5$$

$$\boxed{\text{מסקנה: קיבלנו } c_2 = \tfrac{1}{6}, \; n_0 = 15}$$

Question 3 (20 points)

Consider the following list of functions:

$f_1(n) = \log_2(2^n n^{10})$,

$f_2(n) = n^{\frac{2}{\log_2(n)}}$,

$f_3(n) = 3^{\log_2 n}$,

$f_4(n) = n/5$,

$f_5(n) = \log(n^{10})$,

$f_6(n) = 2^{3n}$,

$f_7(n) = 4^{2021} + 2020$

Arrange the following functions according to the asymptotic ordering (big O), from 'smallest' to 'biggest'. Also, find all pairs of functions for which $f_i = \Theta(f_j)$. Prove all of your claims formally.

$* \; n = 2^{\log_2(n)}$

$x = a^{\log_a(x)}$

$(a^m)^n = a^{mn}$

נכתוב את הפונקציות:

$f_1(n) = \log(2^n \cdot n^{10}) = \log(2^n) + \log(n^{10}) = \boxed{n + 10\log(n)}$

$f_2(n) = n^{\frac{2}{\log_2(n)}} \overset{*}{=} \left(2^{\log(n)}\right)^{\frac{2}{\log n}} = 2^{\log(n)\cdot\frac{2}{\log(n)}} = \boxed{4}$

$f_3 = 3^{\log_2(n)} = 2^{\log\left(3^{\log(n)}\right)} = 2^{\log(n)\cdot\log 3} = \left(2^{\log(n)}\right)^{\log 3} = n^{\log(3)} = \boxed{n^{1.53}}$

$f_4 = \boxed{\frac{n}{5}}$

$f_5 = \log(n^{10}) = \boxed{10\log(n)}$

$f_6 = 2^{3n} = \boxed{8^n}$

$f_7 = \boxed{4^{2021} + 2020}$

פתרון:

$f_1: n+10\log n \le n+10n \le 11n \quad n_0=1 \quad c=11$

$f_2: 4 \le c\cdot 1 \quad c=5$

$f_3: n^{1.53} \le n^{1.53} \quad c=2 \quad n_0\ge 1$

$f_4: \frac{n}{5} \le c\cdot n \quad c=2 \quad n_0>1$

$f_5: 10\log(n) \le c\cdot\log(n) \quad c=12 \quad n_0\ge 2$

$f_6: 8^n \le 8^n\cdot c \quad c=2 \quad n_0\ge 1$

f_1	$n+10\log n$	$O(n)$
f_2	4	$O(1)$
f_3	$n^{1.53}$	$O(n^{1.53})$
f_4	$\frac{n}{5}$	$O(n)$
f_5	$10\log(n)$	$\log(n)$
f_6	8^n	$O(8^n)$
f_7	$4^{2021}+2020$	$O(1)$

$\Downarrow$

$$\boxed{f_{2,7} < f_5 < f_{1,4} < f_3 < f_6}$$

Question 4 (10 points):

Claim: Any group of $n \geq 8$ students, can be partitioned into non-overlapping groups of 4 or 5 students. We'll call it $P(n)$.

Proof:

Base case: For n=8: P(8) - we can divide into 2 groups of 4.

Induction hypothesis:
For any group k≤n, $P(k)$ can be divided into groups of 4 or 5 students.

Induction step:

For P(n+1), we'll form a group of 4 students. Then we can divide the remaining P(n-3) students into groups of 4 or 5 students based on the induction hypothesis.

This proves that for P(n+1), we can divide the students into groups of 4 or 5 students.

What is wrong with this proof?

1. בהנחת האינדוקציה מניחים כי $n \geq k$ ואנו יודעים
אם כי מתקיימת עבור $n \geq 8$ ולכן ההנחה תקפה ותלויה מ 8 ומעלה.
כאשר נבחר $k=8$ באינדוקציה נקבל כי $P(k-3) = P(5)$
אך ההנחה תלויה רק מ $P(n)$ $n \geq 8$. ולכן שגויה.

2. אין מספיק מקרי בסיס, אנו צריכים לכסות מספיק
מקרים מינימליים כך שבזמן החלוקה ל 4 או 5 קבוצות
תמיד נישאר בתחום שאנו יודעים, כמו שהוכח כאן
חסר להוכיח את $\{9,10,12\}$.

$8=4+4$
$9=4+5$
$10=5+5$
$12=4+4+4$

זה יבטיח לנו כי תמיד אנו נוכל
לחסר 4 ועדיין נהיה בתחום. שהוא מגבול מ-8.

3. הטענה לא עבור $n \geq 8$ נכונה ונראה דוגמא נגדית
כי $n=11$ הטענה שקרית ולא ניתן לחלק לקבוצות
של 4 או 5. $\Leftarrow$ $\boxed{n \geq 12}$

$4a+5b=n$

Question 5 (20 points)

For each statement below decide whether it's true or false. Prove your claim using only the definition of $\Theta(\cdot)$. Do not use other properties shown in class.

a. If $f(n)=\Theta(g(n))$, and for every n, $f(n),g(n)\ge 2$ then $\log_2(f(n))=\Theta(\log_2(g(n)))$.
b. If $f(n)=\Theta(g(n))$, and for every n, $f(n),g(n)\ge 2$ then $2^{f(n)}=\Theta(2^{g(n)})$.

:a

1. נתון $f(n)=\Theta(g(n))$ כאשר $f(n),g(n)\ge 2$

ולכן קיימים $c_1,c_2>0$ ו $n\ge n_0$ כך ש

$c_1 g(n)\le f(n)\le c_2 g(n)$

ונראה כי מתקיים $\log(f(n))=\Theta(\log(g(n)))$

~~א~~ $g(n)$ לפי שוויון הקבוע.

O:

1. $\log(f(n))\le \log(g(n))+\log(c_1)$

Ω:

2. $\log(f(n))\ge \log(g(n))+\log(c_2)$

חסם עליון:

$$\log(f(n))\le \log(g(n))\left(1+\frac{\log(c_1)}{\log(g(n))}\right)$$

נבחר n_0 כך שלכל $N\ge n_0$: $\max(\log(c_1),\log(c_2))\cdot 2\le \log(g(n))$

$$\log(f(n))\le \log(g(n))\left(1+\frac{\log(c_1)}{\log(g(n))}\right)\le 2\log(g(n))$$

ומצאנו כי $\boxed{c_1=2}$

הערה: — n_0 כך ש $\log(g(n))$ גדול כי 2 לפחות

מן הקבוע $\log(c_1)$ ולכן לא יכול לעבור את החצי: $\frac{\log c_1}{\log(g(n))}$

Ω:

$$\log(f(n)) \ge \log(g(n)) + \log(c_2) \ge \log(g(n)) - \log(c_2)$$

$$\log(f(n)) \ge \log(g(n))\left(1 - \frac{\log c_2}{\log(g(n))}\right)$$

בחרנו n_0 כך שלכל $n \ge n_0$ $\log(g(n)) \ge \max(\log c_1, \log c_2)\cdot 2$

$\Downarrow$

$$\log(f(n)) \ge \frac{1}{2}\log(g(n))$$

ולכן קיבלנו $\boxed{c_2 = \frac{1}{2}}$

בחרנו את n_0 כך ש $\log(g(n))$ גדול פי 2 לפחות מן הקבוע $\log(c_2)$ ולכן לא יכול לסכור את החצי.

לסיכום:

$$\boxed{2\log(g(n)) \ge \log(f(n)) \ge \frac{1}{2}\log(g(n))}$$

$\boxed{c_1 = 2}$ $\boxed{c_2 = \frac{1}{2}}$

$\boxed{n_0 \le n \text{ כך ש } g(n) \ge 4}$

מקרה 2: $g(n)$ חסומה מלמטה $c \ge g(n) \ge 2$

ולכן גם $f(n)$ חסומה $bc \ge f(n) \ge 2a$ $\log(g(n))$.

$\log(f(n))$ פונק' חסומה וקבועה מספר גודל ולכן

קיימים קבועים $0 < c_1, c_2$, כך שלכל $n_0 \le$

$$\boxed{c_2\log(g(n)) \ge \log(f(n)) \ge c_1\log(g(n))}$$

הטענה נכונה

Question 5 (20 points)

For each statement below decide whether it's true or false. Prove your claim using only the definition of $\Theta(\cdot)$. Do not use other properties shown in class.

a. If $f(n) = \Theta(g(n))$, and for every n, $f(n), g(n) \ge 2$ then $\log_2(f(n)) = \Theta(\log_2(g(n)))$.
b. If $f(n) = \Theta(g(n))$, and for every n, $f(n), g(n) \ge 2$ then $2^{f(n)} = \Theta(2^{g(n)})$.

ב:

נתון כי: $f(n),g(n)\ge 2$ $\quad f(n)=\Theta(g(n))$

ולכן קיימים c_1,c_2 n_0 מתאים

$$c_2 g(n) \le f(n) \le c_1 g(n)$$

נבדוק האם כי $2^{f(n)} = \Theta(2^{g(n)})$

$$c_2 2^{g(n)} \le 2^{f(n)} \le c_1 2^{g(n)}$$

אינטואיציה: בפונקציה הראשונה השינוי היא פנימי.
ובדרך כלל במקרה פונקציה השינוי בקבוע יכנס אל תוך החזקה
ובעצם מכפיל את כולו, משנה את הסדר גודל עצמו.
נבדוק את השערה כשנותן $f(n)=c_i g(n)$
כאשר נבדוק נרצה האם כי היחס מתכנס לקבוע ≠ 0 ולכן תקיים
אם הם לא מתכנס לכן אז חסום עליון.
ניקח את הדוגמא הפשוטה.
כאשר $f(n)=2g(n)$

$$\frac{2^{f(n)}}{2^{g(n)}} = 2^{f(n)-g(n)}$$

$$\boxed{2^{2g(n)-g(n)} = 2^{g(n)} \to \infty}$$

כלומר אין קבועים $c_1,c_2>0$ כך ש $c_1 2^{g(n)} \le 2^{f(n)} \le c_2 2^{g(n)}$.

משל

Question 6 (20 points)

The function *findPair(L,k)*, takes as input a linked list *L* (whose elements are integers) and an integer *k*. The linked list is given as a pointer to the list. The function returns true if and only if there exist two different elements in L with values *x* and *y* such that *x+y = k*. The algorithm shouldn't change *L*, i.e. the linked list *L* is exactly the same during the entire execution of *FindPair*. Moreover, the algorithm should use only a constant number of pointers to list items as an additional memory. Assume that L holds at least two elements and does not contain duplicates.

Write a pseudo code for the function. Write how many comparisons will be performed by the algorithm (that is, how many times the algorithm will check whether two numbers are equal) in the best case and in the worst case as a function of *n*, the length of *L*.

פתרון:

```
findPair(L, k):
    if L.head == null or L.head.next == null:
        return false

    p ← L.head
    while p ≠ null and p.next ≠ null:
        q ← p.next
        while q ≠ null:
            if p.value + q.value == k:
                return true
            q ← q.next
        p ← p.next

    return false
```

הסבר:

האלגו' מקבל רשימה מקושרת ומספר k ובודק האם קיימים שני איברים שונים שסכומם שווה ל k.
אנחנו עוברים עם מצביע p על כל איבר ברשימה ובכל שלב רצים עם מצביע q על שאר האיברים שאחריו ברשימה, בכל פעם אנו בודקים האם

p.val + q.val == k

במידה ומצאנו φ ו ψ שסכומם שווה ל-1 אנו נחזיר
פסוק אמת True. אם סיום לעבור על כל זוג
אפשרי ולא החזיר True במהלך הדרך נחזיר פסוק
שקר שהוא false.
אנו משתמשים בזוג מקבוצים φ, ψ וללא זיכרון
נוסף וללא לשנות את הרשימה.

ניתוח זמן ריצה:

במקרה הטוב אנו ניתקל ב x.ע אשר במקום
הראשון במערך ובמקום שמוק אחרון ואז זמן
הריצה יהיה $\Theta(1)$.

במקרה הגרוע: על האלגוריתם לעבור ולעבור על
כל הקומבינציות ללא חזרות $\Leftarrow$ $\frac{n(n-1)}{2}$ *מעבר על כל הצירוף ללא חזרות
ולכן זמן הריצה יהיה $\Theta(n^2)$

מש"ל