# מבני נתונים - סמסטר ב' תש"פ

## מטלה 2

**הנחיות:**

- מטלה זו הינה להגשה ביחידים. העתקה תגרור **לפסילה מלאה** של המטלה למעתיק והמועתק.
- המטלה מחולקת לשני חלקים: חלק תיאורטי (שאלות 1,2), וחלק מעשי (שאלה 3). עליכם לפתור את החלק התיאורטי ולצרף אותו כקובץ PDF או תמונה. את החלק המעשי יש לצרף כקובץ java בשם Range.java. יש להגיש את המטלה בקובץ ZIP (ולא כל דחיסה אחרת) המכיל את שני הקבצים הנ"ל בלבד. שם קובץ ה- ZIP יהיה מספר ת.ז. של התלמיד. אין להגיש קבצים או תיקיות מיותרים. סטייה מהנחיות אלו תגרור הורדה בציון.

**שאלה 1:**

הניחו שבכל שלב באלגוריתם מיון מהיר, הציר (pivot) בפונקציית העזר partition נבחר כך שמיקומו הסופי מחלק את תת-המערך לשני חלקים; האחד מכיל חלק יחסי של α מאיברי התת-מערך והשני מכיל חלק יחסי של $1-\alpha$ מאיברי התת-מערך, כאשר $0<\alpha\leq 1/2$ הוא קבוע. הראו שהעלה בעל העומק המינימלי בעץ הריקורסיה (זאת אומרת, העלה המתאים למסלול הקצר ביותר המתחיל בשורש העץ ומסתיים בעלה) הוא בקירוב בעומק $\frac{\log n}{\log(1/\alpha)}$ ועומק עץ הריקורסיה (זאת אומרת, עומק העלה המתאים למסלול הארוך ביותר המתחיל בשורש העץ ומסתיים בעלה) הוא בקירוב $\frac{\log n}{\log(1/(1-\alpha))}$. על מנת לקבל את הקירובים הנ"ל התעלמו מעיגול המספרים לשלמים.

**שאלה 2:**

ראינו בשיעור שכל אלגוריתם מיון (נכון) מבוסס השוואות נדרש ל- $\Omega(n\log n)$ פעולות במקרה הגרוע ביותר, שכן מספר התשובות השונות שעליו להבדיל ביניהן הוא כמספר הפרמוטציות, $n!$. הראו שלא קיים אלגוריתם מיון (נכון) מבוסס השוואות שמשיג זמן ריצה **לינארי** $O(n)$ עבור $m\geq n!/2^n$ מהקלטים האפשריים באורך $n$ (כאשר $n$ מספיק גדול). ניתן להשתמש ללא הוכחה בעובדות הבאות:

- מספר העלים המקסימלי עד לעומק $h=O(n)$ בעץ בינארי הוא לכל היותר $2^{h-1}=2^{O(n)}$.
- מתקיים אי-השוויון $\log(n!)\geq\frac{n}{2}\log\frac{n}{2}$.

**שאלה 3:**

ממשו בשפת Java מחלקה בשם Range.java שבהנתן $n$ מספרים שלמים בטווח 0 עד $k$,
ראשית מבצע (בבנאי / constructor) עיבוד מקדים של הקלט בזמן ריצה $\Theta(n+k)$,

**public Range(int[] a, int k)**

ולאחר מכן עונה בזמן ריצה $O(1)$ על כל שאילתא מן הצורה: בהנתן מספרים שלמים
$0 \le a \le b \le k$, כמה מספרים מתוך ה- $n$ נופלים בטווח $[a \dots b]$?

**public int query(int a, int b)**

רמז: השתמשו בוריאציה מתאימה של אלגוריתם מיון מנייה (הידוע גם כמיון בטווח קצוב, counting sort, או bounded sort) עבור שלב העיבוד המקדים. שימו לב שזמן הריצה של פונקציית **query** חייבת להיות $O(1)$ ולא $O(k)$

# ב ה צ ל ח ה !