**גרפים אקראיים במודל ארדש – ריניה**

מגישים:

יאיר זוהר, 208959544  
איתי בלייכר, 208632752

תיאור המטלה:

במטלה זו מימשנו פונקציה אשר יוצרת גרפים אקראיים ולאחר מכן חקרנו 3 תכונות אצל כל גרף.

שלושת התכונות שחקרנו הם:  
תכונה 1 – קשירות הגרף, האם הגרף קשיר או לא?  
תכונה 2 – קוטר הגרף, האם קוטר הגרף שווה ל-2 או גדול ממנו?  
תכונה 3 – האם קיימת צומת מבודדת בגרף?

כל גרף בתרגיל מורכב מ-1000 צמתים וכל תכונה נבדקה 500 פעמים על גבי 10 הסתברויות שונות (בסה"כ נבדקו במהלך המטלה 15,000 גרפים).

אנו בחרנו לייצג את הגרף במטלה ע"י שימוש במערך שייצג את כמות הצמתים בגרף (במקרה שלנו מערך עם 1000 תאים, אך כמות זו היא כמובן דינאמית וניתנת לשינוי לפי צורך המשתמש), כל תא מצביע על רשימה מקושרת אשר מסמלת את הצמתים השכנים לאותו תא (במילים אחרות הצמתים אשר מחוברים לצומת הזו בעזרת צלע).  
ע"י שימוש במבנה נתונים זה ניתן לייעל בצורה מאוד משמעותית את זמן הריצה של התוכנית מכיוון שצלעות אשר לא קיימות לא מיוצגות (בניגוד למטריצה אשר בה הן כן מיוצגות).

הפונקציות המרכזיות אשר כתבתנו עבור ביצוע המטלה הן:

double rnd\_gnr() – הגרלת מספר בין 0 ל-1  
typedef struct Queue – מבנה נתונים של תור עם פונקציות הכנסה, הוצאה ושחרור זיכרון  
typedef struct AdjListNode – תא ברשימה המקושרת  
typedef struct AdjList – מבנה נתונים של רשימה מקושרת  
typedef struct Graph - מבנה נתונים של גרף  
AdjListNode\* newAdjListNode(int dest) – הוספת תא לרשימה המקושרת  
void addEdge(Graph\* graph, int src, int dest) – הוספת צלע לגרף  
Graph\* build\_random\_graph(int V, float P) – בניית גרף רנדומלי  
void printGraph(Graph\* graph) – הדפסת גרף  
void freeGraph(Graph\* graph) – שחרור הזיכרון של הגרף  
int is\_isolated(Graph\* graph) – בדיקת תכונה 3 על גרף ספציפי  
int BFS(Graph\* graph, int src, bool\* visited, int\* distance) – BFS אלגוריתם   
int connectivity(Graph\* graph) – בדיקת תכונה 1 על גרף ספציפי  
int diameter(Graph\* graph) – בדיקת תכונה 2 על גרף ספציפי  
void threshold1(int V, int NOGraphs) – חקירת תכונה 1 עם 10 הסתברויות שונות  
void threshold2(int V, int NOGraphs) – חקירת תכונה 2 עם 10 הסתברויות שונות  
void threshold3(int V, int NOGraphs) – חקירת תכונה 3 עם 10 הסתברויות שונות

\* תיעוד והסבר מפורט על כל פונקציה נמצא בקובץ הקוד.

תוצאות הסימולציות:





