**מציאת מינימום ומקסימום במערך-** **minMax**

**maxMaxM2Recursive - - 3n/2 השוואות**

**הסבר פתרון –**

* תחילה נבדוק את גודל המערך, אם גודל המערך שווה ל-0 אז אין לנו איברים במערך ואז אנחנו נחזיר 0.
* אם לא נבדוק האם גודל המערך הוא 1, במידה שכן נגדיר את המקסימום1 והמקסימום2 להיות האיבר היחיד שקיים לנו במערך.
* במידה והבדיקות האלה לא התקיימו נגדיר משנה (0=comparisons) שמייצג לנו את כמות ההשוואות שאנחנו עושים בפונקציה, משתנה Max שמייצג את האיבר המקסימלי במערך ומשתנה Max2 שמייצג את האיבר המקסימאלי השני (הקטן מMax).
* תחילה נגדיר מערך של nodes בגודל המערך המקורי נרוץ על הגודל שלו וניצור כל פעם Node חדש עם הערך שיש במערך (כל אובייקט של node מחזיק בתוכו משתנה שמחזיק את הערך שלו מהמערך המקורי ומשתנה של stack שנחזיק בו את כל האופציות האפשריות להיות המקסימום השני [כל האיברים שקטנים ממנו]).
* לאחר מכן נקרא לפונקציה רקורסיבית שתחזיר לנו את הnode המקסימלי (שמכיל גם את הערך וגם את הstack).
* הפונקציה היא בעצם MeregSort בעזרת חלוקת המערך לשני חלקים שווים עד שמגיעים ליחידות ומיחידות אלו ליצור זוגות כאשר נכנסנו לפונקציה בפעם הראשונה נבדוק תנאי עצירה (האם האינדקסים שווים) במידה וכן נחזיר את הnode במיקום low במערך, לאחר מכן ניקח את הממוצע של low ו-high ואז נקרא לפונקציה הרקורסיבית ונעביר לה את ההתחלה ואת האמצע (צד ימין), אחרי זה נקרא שוב לפונקציה הרקורסיבית ונעביר לה את האמצע + 1 וhigh- לאחר שהקריאות לפונקציה הרקורסיבית הסתיימו אנחנו חוזרים למקרה הראשון ונבדוק האם הערך של צד ימין גדול מצד שמאל לכן נוסיף 1 ל-comparisons בכל בדיקה.
* במידה וכן נוסף את הערך של צד שמאל למחסנית של קודקוד ימין ונחזיר את ה-node השמאלי.
* במידה ולא נוסף את הערך של צד ימין למחסנית של קודקוד שמאל ונחזיר את ה-node הימיני.

לאחר מכן נוציא את האיבר שנוסף אחרון מהמחסנית של הקודקוד שחזר לנו מהפונקציה ונכניס אותו למשתנה x, נבצע השמה לMax להיות שווה לערך של הקודקוד שחזר לנו מהפונקציה ונבצע השמה גם לMax2 להיות שווה ל-x.

* לאחר מכן נרוץ כל עוד המחסנית של הקודקוד המקסימלי אינו ריק, ובכל ריצה נוציא את האיבר שנוסף אחרון למחסנית ונבצע השוואה בין Max2 לבין הערך שאנחנו מוציאים מהמחסנית לכן נוסיף 1 ל-comparisons, במידה והערך במחסנית גדול מהאיבר Max2 נבצע השמה Max2 להיות שווה לאיבר הזה.
* בסוף נחזיר את המשתנה comparisons

**הוכחת סיבוכיות-**

הפונקצית עזר שלנו היא בעצם MeregSort בעזרת חלוקת המערך לשני חלקים שווים עד שמגיעים ליחידות ומיחידות אלו ליצור זוגות, ככה קיבלנו תהליך רקורסיבי, בסוף התהליך נקבל אינדקס של איבר מקסימלי Max ומחסנית ששייכת אליו ומכילה את Max2.  
מספר ההשוואות: בפעם הראשונה מספר ההשוואות היא בפעם השנייה וכן הלאה.  
סה"כ   
כאשר  *הוא מספר איברי המערך, בגלל שאנחנו כל פעם מחלקים מערך ל-2 חלקים שווים, מספר הכניסות למחסנית הוא ולכן מספר ההשוואות הכולל הוא   
ניתן לממש את האלגוריתם בצורה רקורסיבית כוון שגובה העץ הרקורסיה הוא*

**minMaxM2 - - 3n/2 – 3 + 2 השוואות**

**הסבר פתרון –**

* תחילה נבדוק את גודל המערך, אם גודל המערך שווה ל-0 אז אין לנו איברים במערך ואז אנחנו נחזיר 0.
* אם לא נבדוק האם גודל המערך הוא 1, במידה שכן נגדיר את המקסימום1 והמקסימום2 להיות האיבר היחיד שקיים לנו במערך.
* במידה והבדיקות האלה לא התקיימו נגדיר משנה (comparisons = 0) שמייצג לנו את כמות ההשוואות שאנחנו עושים בפונקציה, נשלח את המערך לפונקציית עזר שתבדוק לנו את 2 האיברים הראשונים, ולכן נעלה את comparisons ב-1 כיוון שיש לנו השוואה אחת, הפונקציה בודקת מי יותר גדול ומגדירה את המקסימאלי להיות בMax ואת המקסימאלי השני(המקסימלי שקטן מMax) להיות בMax2.
* נרוץ על המערך בעזרת משתנה i ממיקום 2 ועד סוף המערך (בכל ריצה נקדם את i ב-2), בכל ריצה נבדוק מי יותר גדול מכל 2 האיברים הבאים במערך (בריצה הראשונה נבדוק את arr[2]< arr[3]) נעלה את comparisons ב-1 מכיוון שיש לנו את ההשוואה לבדיקת מקסימום בין שני האיברים והמקרה הזה מתחלק לשני מקרים:

**מקרה א:** אם [arr[2] < arr[3 מתקיים אנחנו צריכים לבצע עוד 2 בדיקות, הראשונה האם arr[3] > Max ולכן נעלה את comparisons ב-1, במידה וזה מתקיים נשנה את האיבר המקסימלי השני (Max2) להיות Max ואת האיבר המקסימלי(Max) להיות האיבר במיקום i+1 arr[3].

לאחר מכן נבדוק האם arr[2] < Max2ולכן נעלה את comparisons ב-1, במידה וזה מתקיים נשנה את האיבר המקסימלי השני (Max2) להיות האיבר במיקום i arr[2] .

**מקרה ב:** אם התנאי של מקרה א לא התקיים זה אומר ש: [arr[2] > arr[3 מתקיים אנחנו צריכים לבצע עוד 2 בדיקות, הראשונה האם arr[2] > Max ולכן נעלה את comparisons ב-1, במידה וזה מתקיים נשנה את האיבר המקסימלי השני (Max2) להיות Max ואת האיבר המקסימלי(Max) להיות האיבר במיקום i arr[2]

לאחר מכן נבדוק האם arr[3] > Max2ולכן נעלה את comparisons ב-1, במידה וזה מתקיים נשנה את האיבר המקסימאלי השני להיות האיבר במיקום (i+1) arr[3] .

* לכן בכל הריצות האלה יש לנו סה"כ 3 השוואות (1 בין כל שני איברים, 1 בין האיבר הגדול למקסימום ו-1 בין האיבר השני למקסימום השני) וכמות הריצות היא גודל המערך חלקי 2.
* לאחר מכן נבדוק נבדוק האם גודל המערך שלנו הוא זוגי:
* במידה והוא זוגי: אז סיימנו את האלגוריתם, ונחזיר את המשתנה comparisons .
* במידה והוא אי-זוגי: נבדוק האם האיבר במיקום האחרון במערך גדול מהאיבר המקסימלי (Max) לכן נעלה את comparisons ב-1
* אם התנאי מתקיים את נשנה את האיבר המקסימלי השני (Max2) להיות האיבר (Max) ואת האיבר (Max2) להיות האיבר במיקום האחרון במערך (גודל המערך פחות 1).
* אם התנאי אינו מתקיים נבצע עוד השוואה ונבדוק האם האיבר במיקום האחרון במערך גדול מהאיבר המקסימאלי השני (Max2), לכן נעלה את comparisons ב-1, במידה והתנאי כן מתקיים נשנה את האיבר המקסימאלי השני(Max2) לאיבר שנמצא במיקום האחרון במערך (גודל המערך פחות 1).

**הוכחת סיבוכיות-**

אנחנו רצים על המערך וכל פעם מקדמים את האינדקס ב2 לכן אנחנו רצים סה"כ בכל ריצה אנחנו מבצעים 3 השוואות (חוץ מהריצה הראשונה שבדקנו אותה לפני הלולאה) (1 בין כל שני איברים סמוכים, 1 בין האיבר המקסימלי מתוך השניים ל(Max), 1 בין האיבר המינימאלי מתוך השניים שבדקנו ל-(Max2)) ולכן בכל ריצה יש לנו 3 השוואות.

הסיבוכיות מתחלקת ל-2

1. המקרה שגודל המערך הוא זוגי כלומר, כלומר יהיו לנו 3n/2 השוואות פחות הבדיקה של שני האיברים הראשונים במערך.
2. המקרה שגודל המערך הוא אי-זוגי כלומר:  
   ה-2 זה ההשוואות של האיבר האחרון

הסיבוכיות היא: