**שאלה 3 – אלגוריתמיקה**

Binary search זהו אלגוריתם יעיל יותר לחיפוש איבר בתוך מערך. החיפוש יכול להתבצע רק על ידי מערך ממוין. (בדוג' הוא יהיה ממוין בסדר עולה)

נניח אנחנו מחפשים את X. במהלך החיפוש, בכל שלב אנחנו נחלק את המערך ל2 מערכים ע"י הכרזה על משתנים min, max, mid כאשר min=0, max=arr.length, mid=(min+max)/2.

לאחר מכן, נבדוק אם X שווה לערך הגדול ביותר בחצי הקטן של המערך (בין min לmid). במידה ולא, נבדוק אם הוא גדול מהערך הגדול ביותר במערך הקטן. במידה והוא יותר גדול, אז נשאיר בצד את המערך הקטן ונמשיך עם הגדול בשני (כלומר, min=mid, mid=(min+max)/2; כי אם X קיים במערך, אז הוא יהיה במערך הגדול) במידה והוא קטן מהערך הגדול ביותר של המערך הקטן, אז נמשיך עם המערך הקטן ונשאיר בצד את הגדול (על אותו עיקרון כמו באפשרות הראשונה). ככה אנחנו מקטינים כל פעם את כמות האיברים בחצי.

נמשיך עם החלוקות שוב ושוב עד שמוצאים את האיבר X או עד שהאינדקס של min יהיה גדול מהאינדקס של max. במצב כזה אפשר להבין כי נגמרו כל האפשרויות וX אינו במערך.

**בונוס:**

1. מצב שבו האלגוריתם יכול להיכשל או לתת תוצאות לא נכונות הוא כאשר המערך שעליו מפעילים את האלגוריתם לא יהיה ממוין. כדי לפתור מצב כזה, יש צורך קודם למיין את המערך ורק לאחר מכן להשתמש בו.
2. הסיבוכיות של Binary Search הוא (O(log n) ולכן הוא יעיל יותר מLinear Search Interpolation Search או משום שהם עם סיבוכיות של (O(n)). במצב שבו המערך לא ממוין, עדיף להשתמש בlinear search משום שמיון המערך הוא בעל סיבוכיות של (O(n^2) ולכן השימוש בLinear Search הוא יותר יעיל. במצב שבו המערך ממוין אך לא בקפיצות אחידות, לא ניתן להשתמש בInterpolation Search ולכן יהיה יעיל ביותר להשתמש בBinary Search. במצב שבו המערך ממוין בעל קפיצות שוות, הInterpolation Search הוא היעיל מבין שלושתם כי יעילותו היא (O(log log n)).
3. גרסה מקבילית לBinary Search שעשויה להוריד את רמת הסיבוכיות של האלגוריתם היא להכניס פרמטר נוסף (לחלק את mid לmid1, mid2) ובכך בכל פעם יהיו לנו 3 מערכים לבדיקה. כך, ב2 תנאים יהיה ניתן להקטין ב2/3 את המערך (במקרה הטוב). יחד עם זאת, במקרה הגרוע זה לא יקל על הסיבוכיות