

שאלה 3: הסבר על אלגוריתם Binary Search

מהו Binary Search?

Binary Search הוא אלגוריתם יעיל למציאת איבר מסוים במערך שממויין מראש. האלגוריתם עובד על עיקרון של חלוקת הטווח שבו מחפשים כל פעם לחצי, וכך מצמצם משמעותית את מספר הבדיקות הנדרשות.

איך האלגוריתם פועל?

- תחילה בוחנים את האיבר האמצעי במערך.
- אם האיבר האמצעי הוא הערך המבוקש, סיימו.
- אם הערך המבוקש קטן מהאיבר האמצעי, ממשיכים לחפש רק במחצית השמאלית של המערך.
- אם הערך המבוקש גדול מהאיבר האמצעי, ממשיכים לחפש במחצית הימנית.
- חוזרים על התהליך עד שמוצאים את הערך או שהטווח ריק, כלומר האיבר אינו נמצא במערך.

סיבוכיות זמן

החיפוש מצמצם את טווח החיפוש בחצי בכל איטרציה, לכן מספר הצעדים המקסימלי הדרוש הוא לוגריתמי ביחס לגודל המערך, כלומר $O(\log n)$ סיבוכיות זו הופכת את Binary Search לאלגוריתם מהיר מאוד, גם במערכים גדולים.

בנוס 1: מתי החיפוש עלול להיכשל?

החיפוש הבינארי מחייב שהמערך יהיה ממוין. אם המערך אינו ממוין, התהליך לא יוכל לקבוע באיזו מחצית להמשיך לחפש, ולכן התוצאות עלולות להיות שגויות או שהחיפוש ייכשל. כדי להתמודד עם זאת, יש לוודא שהמערך ממוין לפני הרצת החיפוש. אם המערך לא ממוין, יש למיין אותו קודם באמצעות אלגוריתמי מיון כמו Merge Sort או Quick Sort. אם לא ניתן למיין, ניתן להשתמש באלגוריתם חיפוש לינארי, שהוא איטי יותר אבל מתאים למערכים לא ממוינים.

בנוס 2: השוואה בין אלגוריתמים נפוצים

חיפוש לינארי עובר על כל איבר במערך אחד-אחד ולכן סיבוכיותו היא $O(n)$. הוא מתאים בעיקר למערכים קטנים או לא ממוינים. חיפוש בינארי מתאים למערכים ממוינים ומאופיין בסיבוכיות $O(\log n)$, מה שמאפשר חיפוש מהיר יותר.

חיפוש אינטרפולציה הוא שיפור על חיפוש בינארי במקרים בהם הנתונים מפולגים באופן אחיד, למשל ערכי מספרים ברצף, והוא עובד בממוצע בזמן טוב יותר, בסביבות $O(\log \log n)$. זה לא מספר קבוע והוא גם יכול להגיע לזמן לינארי כי הוא מסתמך על רציפות בהפרשים.

בונוס 3: גרסה מקבילית של Binary Search

במערכות עם מעבדים מרובי ליבות ניתן לשפר את הביצועים של חיפוש בינארי על ידי ביצוע חיפושים במקביל. ישנן שתי שיטות עיקריות לחלוקת המשימות:

1. חלוקה סטטית של הטווחים:

מחלקים את המערך למקטעים נפרדים, וכל ליבה אחראית על חיפוש בתוך הטווח שהוקצה לה. כל ליבה מבצעת חיפוש בינארי עצמאי על טווח קטן יותר. אם אחת הליבות מוצאת את האיבר, מתבצעת הפסקת חיפוש עבור כל הליבות האחרות.

2. סנכרון דינמי של הטווחים:

בהתחלה כל הליבות מחפשות במקביל, אבל ברגע שברור שהטווח הצריך חיפוש מחוץ לטווח של ליבה אחת וקטן מהטווח של ליבה אחרת, כל הליבות מעדכנות את הטווח החדש ומשתפות פעולה, כלומר כולן מחפשות את אותו הטווח המצומצם. כך מתאפשר סנכרון חכם ויעיל, תוך ניצול מקסימלי של כל הליבות גם בטווחים קטנים.

שיטה זו מאפשרת ניצול טוב יותר של כוח המחשוב, אך מצריכה תיאום וסנכרון בין הליבות כדי להימנע מביצוע חיפושים מיותרים, מה שעלול לגרום להאטות בגלל תקשורת בין הליבות.