בעיית עץ החיפוש הבינארי האופטימלי

n ערכים לחיפוש $k_1, ...k_n$. n הערכים ממויינים: $k_1 < k_2 < ... < k_n$
ייתכנו מילים $k_i < d_i < k_{i+1}$. נרצה לסמן את זה בעץ.
לכל ערך יש הסתברות שנחפש אותו. לערך k_i ההסתברות היא p_i. לכל d_i הסתברות שנחפש מילה בטווח של d_u היא q_i.

$$\sum_{i=1}^{n} p_i + \sum_{i=0}^{n} q_i = 1$$

המטרה: לתכנן עץ חיפוש עם תוחלת זמן חיפוש מינימלית.
($D_T[*]$ - העומק של * בעץ T. נגדיר שעומק השורש הוא 0)
בעץ T תוחלת זמן החיפוש היא:

$$E\left[seek\, time\right] = \sum_{i=1}^{n} p_i \left(D_T\left[k_i\right] + 1\right) + \sum_{i=0}^{n} q_i \left(D_T\left[d_i\right] + 1\right)$$

אלגוריתם נאיבי

עוברים על כל האפשרויות לעצים עם n ערכים. מס' האפשרויות הוא מספר קטלן $\Omega\left(\frac{4^n}{n^{1.5}}\right)$

אלגוריתם רקורסיבי

בהינתן קטע $k_i, ...k_j$, נרצה למצוא את המילה שאם נבחר אותה בתור שורש החיפוש יהיה אופטימלי, כאשר כל אפשרות מחלקת את הקטע לשני קטעים, שאותם אפשר לבדוק ברקורסיה.

נעבור על כל האפשרויות לשורש של העץ. לכל אפשרות נחשב רקורסיבית את תת העץ השמאלי האופטימלי ואת תת העץ הימני האופטימלי, ואז נבחר את האפשרות לשורש שנותנת תוחלת זמן מינימלית.

נגדיר $e\left(i, j\right)$ תוחלת זמן החיפוש האופטימלית לעץ שהערכים בו הם $k_i...k_j$

$$d_{i-1} < k_i, ... < k_j < d_j$$

$$e\left(i, j\right) = \min_{i \le r \le j} \left\{e\left(i, r-1\right) + e\left(r+1, j\right)\right\} + w\left(i, j\right)$$

$$w\left(i, j\right) = \sum_{k=i}^{j} p_k + \sum_{k=i-1}^{j} q_k$$

$$\forall_{i \le r \le j} w\left(i, j\right) = w\left(i, r-1\right) + p_r + w\left(r+1, j\right)$$

$w\left(i,j\right)$- ההסתברות שנגיע לקודקוד בתת העץ מk_i עד לk_j.
תנאי עצירה - $e\left(i,i-1\right)=q_{i-1}$
זמן ריצה של הפתרון הרקורסיבי:

$$T\left(n\right)=\sum_{i=1}^{n}\left(T\left(n-i\right)+T\left(i\right)\right)+w\left(i,j\right)-\delta\geq 2T\left(n-1\right)+\theta\left(1\right)\rightarrow\Omega\left(2^{n}\right)$$

קיבלנו סדר גודל אקספוננציאלי, שכן אנו מחשבים תתי בעיות שוב ושוב. עדיף להשתמש בתכנות דינאמי.

אלגוריתם תכנות דינאמי

Optimal BST($\left\{p_i\right\},,\left\{q_i\right\},n$)

1. for i←1 to n+1
   - 1.1 E[i,i-1]← q_{i-1}
   - 1.2 w[i,i-1]← q_{i-1}
2. for L←1 to n
   - 2.1 for i← 1 to n-L+1
     - 2.1.1 j←i+L-1
     - 2.1.2 E[i,j]← ∞
     - 2.1.3 w[i,j]←w[i,j-1]+p_i+q_j
     - 2.1.4 for r←i to j
       - 2.1.4.1 t←E[i,r-1]+E[r+1,j]+w[i,j]
       - 2.1.4.2 if t<E[i,j] then
         - 2.1.4.2.1 E[i,j]←t
         - 2.1.4.2.2

מבנה הנתונים: מטריצה $E\left[i,j\right]$, מטריצה $w\left[i,j\right]$, ומטריצת הפיתרון $root\left[i,j\right]$

נכונות הנוסחא הרקורסיבית

האלגוריתם מחשב נכון את $E\left[i,j\right]$ עבור תת העץ $k_i,...k_j$:
באינדוקציה על מספר הערכים בעץ m. עבור $m=1$ יש רק אפשרות אחת, ולכן הוא בהכרח אופטימלי.
הנחת האינדוקציה: נניח נכונות לתתי עצים מגודל קטן מm.
צעד: נוכיח לתת עץ בגודל m.
נסתכל על עץ T של המפתחות $k_i,...k_j$, $j-i=m-1$. יש לו עלות $E\left[i,j\right]$. נניח בשלילה שיש עץ T' בעל תוחלת קטנה מזו של T. נסתכל על r' - השורש בT'. יש לו תת עץ ימני T'_R ותת עץ שמאלי T'_L.

$$E'=E\left[T'_R\right]+E\left[T'_L\right]+w\left(i,j\right)$$

ידוע מהנחת האינדוקציה $E\left[T_R\right]\le E\left[T'_R\right]$ וגם $E\left[T_L\right]\le E\left[T'_L\right]$

$$E\left[T\right]=E\left[i,j\right]=\min_{i\le r\le j}\left\{E\left[i,r-1\right]+E\left[r+1,j\right]\right\}+w\left(i,j\right)$$

$$=E\left[T_L\right]+E\left[T_R\right]+w\left(i,j\right)\le E\left[T'_L\right]+E\left[T'_R\right]+w\left(i,j\right)=E\left[T'\right]$$

קיבלנו $E\left[T\right]\le E\left[T'\right]$ $\Leftarrow$ סתירה

בעיית חלוקת מערך לקטעים

נתון מערך עם n איברים. k-חלוקה של המערך היא חלוקה לk קטעים לא ריקים.
עלות של חלוקה: סכום המכפלות של כל הערכים בכל תת-קטע.

המטרה: למצוא k-חלוקה בעלת ערך מקסימלי.

נגדיר: $N_{i,j}=\prod_{t=i}^{j}A\left[t\right]$

נגדיר $f\left(i,j\right)$ עלות מקסימלית של j-חלוקה של המערך ממיקום 1 עד i.

$$f\left(i,j\right)=\max_{j\le r\le i}\left\{f\left(r-1,j-1\right)+N_{r,i}\right\}$$

תנאי עצירה: N_1 $f\left(i,1\right)$. אם $j<i$ אז $f\left(i,j\right)=0$

הוכחת הנוסחה באינדוקציה על j

עבור $j=1$ נוכיח את $\forall_i f\left(i,1\right)$

נניח עבור ערכים קטנים מk

נוכיח עבור $\forall_i f\left(i,k\right)$