פתרון שאלה 4 בממ"ן 12 (בעיה 6-2 מהספר)

א. הערמה מיוצגת ע"י מערך בצורה דומה לערמה בינארית. (הצמתים מופיעים במערך עפ"י סדר הרמות ובכל רמה עפ"י הסדר משמאל לימין.)

טענה: הבן ה-j של האיבר הנמצא באינדקס i נמצא באינדקס: $son(i, j) = d(i – 1) + j + 1$

נכונות: עבור בניו של האיבר הראשון זה ברור (הם נמצאים במקומות 2 עד $d + 1$).

נניח נכונות עבור $k \le i$ ונראה שמכך נובעת נכונות עבור $k = i + 1$.

עפ"י ההנחה, בנו האחרון של האיבר ה-i נמצא באינדקס $d \cdot i + 1$ ועל כן בנו הראשון של האיבר ה- $i + 1$ נמצא באינדקס $d{\cdot}i + 2$, בהתאם לנוסחה.

בהינתן האינדקס i של צומת כלשהו, האינדקס של אביו של i הוא: $\lceil (i – 1) / d \rceil$.

(כדי להוכיח נכונות מספיק לבדוק, שמהבנים הראשון והאחרון של הצומת שבאינדקס i מגיעים בעזרת הנוסחה חזרה ל-i.)

ב. בעץ שלם בגובה h יש $(d^{h+1} – 1) / (d –1)$ איברים (סכום סדרה הנדסית). לכן, בערמה בגובה h יש בין $(d^{h+1} – 1) / (d – 1)$ איברים (עץ שלם בגובה h) לבין $(d^h – 1)/(d – 1) + 1$ איברים. (עץ שלם בגובה $h – 1$ ועוד איבר אחד).

כלומר, מצד אחד: $n \le (d^{h+1} – 1) / (d – 1) < (d^{h+1}) / (d – 1)$ ולכן $h > \log_d(n(d – 1)) –1$

מצד שני: $n \ge (d^h – 1) / (d – 1) + 1 = ((d^h – 1) + (d – 1)) / (d – 1) \ge (d^h – 1) / (d – 1)$

(באי-שוויון האחרון השתמשנו בעובדה ש- $d \ge 2$) ולכן $h \le \log_d(n(d – 1))$

לפיכך, h הוא המספר השלם הקטן או שווה ל- $\log_d(n(d – 1))$, **הקרוב ביותר** ל- $\log_d(n(d – 1))$

(כי ההפרש ביניהם **קטן ממש** מ-1). מכך נובע ש- $h = \lfloor \log_d(n(d – 1)) \rfloor$.

ג. השגרה EXTRACT-MAX בנויה בצורה הבאה: האיבר המכסימלי שיושב בראש העץ מוצא ומוחלף בעלה האחרון. עתה ייתכן שהופרה תכונת הערמה ושראש העץ קטן מ(לפחות אחד) מבניו. כדי לתקן זאת קוראים ל-MAX-HEAPIFY. (לכן בפרוצדורה EXTRACT-MAX עצמה אין הבדל בין ערמה d-ית לערמה רגילה.)

השגרה MAX-HEAPIFY בודקת אם השורש קטן מ(לפחות אחד) מבניו, ואם כן – מחליפה אותו בבן המכסימלי. לכן בערמה d-ית יש לבדוק את ערך השורש כנגד כל d הבנים (סיבוכיות $O(d)$).

סיבוכיות הזמן: בשגרה עצמה מתבצעת $O(1)$ עבודה. השגרה MAX-HEAPIFY מבצעת $O(d)$ עבודה בכל קריאה ועשויה להיקרא לכל רמה בעץ (שגובהו, כפי שראינו, $\lfloor \log_d(n(d – 1)) \rfloor$).

לכן סה"כ סיבוכיות הזמן היא $O(d{\cdot}\log_d n)$.

הערה: עדיף לא להניח ש-d קבוע. ייתכן שבאפליקציה כלשהי d תלוי בגודל הקלט.

ד. השגרה INSERT מגדילה ב-1 את גודל הערמה ומציבה באיבר החדש $–\infty$. לאחר מכן היא קוראת לשגרה HEAP-INCREASE-KEY עם הפרמטרים המתאימים. מכיוון שגם בערמה d-ית יש לכל איבר אב אחד, אין כל הבדל בין השגרה INSERT בערמה d-ית לבין השגרה INSERT בערמה בינארית (מלבד נוסחה שונה לחישוב מקום האב).

סיבוכיות הזמן: $O(\log_d n)$.

ה. HEAP-INCREASE-KEY (A, i, k): ראשית, השגרה תשווה בין A[i] לבין k. אם $k \le A[i]$, אז לא צריך לעשות דבר. אחרת, השגרה תציב את הערך k ב-A[i]. תכונת הערמה עלולה להיות מופרת, משום שעתה ייתכן ש-A[i] גדול מאביו. במקרה זה יש להשתמש באותה לולאה שמחליפה את A[i] עם אבותיו, כמו בשגרה HEAP-INCREASE-KEY הרגילה.

סיבוכיות הזמן: $O(\log_d n)$, כמו השגרה HEAP-INCREASE-KEY הרגילה.