

3 חלק רטוב

3.1 מימוש מבנה נתונים גנרי – GDT (Generic Data Type)

בחלק זה נממש ADT גנרי בעבור מילון ממוין. קובץ הממשק map.h נמצא בתיקיית התרגיל. עליכם לכתוב את הקובץ map.c המממש את מבנה הנתונים המתואר.

מאחר והמילון הממוין גנרי, יש לאתחל אותו עם מספר מצביעים לפונקציות אשר יגדירו את אופן הטיפול באיברים המאוחסנים בו.

כדי לאפשר למשתמשים במילון (לא לכם!) לעבור על איבריו סדרתית, לכל מילון מוגדר איטרטור (מלשון איטרציה, מעבר על איברים) פנימי יחיד שבעזרתו יוכל המשתמש לעבור על כל איברי המילון.

האיטרציה על איברי המילון צריכה להבטיח למשתמש מעבר על האיברים בסדר עולה מהמפתח הקטן למפתח הגדול – נדע לעשות זאת באמצעות פונקציית השוואת מפתחות (keyCompare) שמסופקת בעת יצירת המילון. פונקציית compare מחזירה 0 אם שני המפתחות שהיא מקבלת שווים, ערך חיובי אם המפתח הראשון גדול מהשני, וערך שלילי אם המפתח השני גדול מהראשון (בדומה לstrcmp).

כדי לבדוק את התנהגות המילון, מסופק סטט בסיסי בקובץ map_example_test.c. להלן הגדרת הפעולות על מילון שעליכם לממש:

1. **mapCreate** – יצירת מילון ממוין חדש. בפעולה זו מוגדרות לרשימה הפעולות בעזרתן ניתן להעתיק, לשחרר מפתחות וערכים ולהשוות מפתחות.
2. **mapDestroy** – מחיקת מילון קיים תוך שחרור מסודר של כל הזיכרון שבשימוש.
3. **mapCopy** – העתקת מילון קיים לעותק חדש כולל העתקת האיברים עצמם (מפתחות וערכים).
4. **mapGetSize** – החזרת מספר המפתחות במילון.
5. **mapContains** – יוחזר true אם המפתח הנתון קיים במילון, אחרת יוחזר false.
6. **mapPut** – שינוי ערך של מפתח קיים או הוספת זוג מפתח-ערך חדש למילון.
7. **mapGet** – החזרת הערך הממופה למפתח הנתון – לא לשכפל אותו (כלומר אין להשתמש בפונקציית העתקת איבר, אלא להחזיר את המצביע לערכו).
8. **mapRemove** – מחיקת איבר מהמילון.
9. **mapGetFirst** – הזזת האיטרטור לתחילת המילון והחזרת עותק של המפתח הראשון. ניתן להניח כי לא תהיה הקצאה שנכשלה בעת החזרת העותק.
10. **mapGetNext** – קידום האיטרטור והחזרת עותק של המפתח המוצבע על ידו. ניתן להניח כי לא תהיה הקצאה שנכשלה בעת החזרת העותק.
11. **mapClear** – ריקון המילון (בשונה מmapDestroy שגם משחרר את כל הזיכרון שבשימוש המילון).

3.1.1 דגשים נוספים ודרישות מימוש

- קראו את התיעוד ב-map.h! הוא מגדיר במפורש כל פעולה שעליכם לממש ויעזור לכם במיוחד!
- קיימות פונקציות שלהן מספר ערכי שגיאה אפשריים. בהערה מעל כל פונקציה תוכלו למצוא את כל השגיאות שיכולות להתרחש בעת קריאה אליה בקובץ הממשק שמסופק לכם (מתחת למילה return בהערה). במקרה של כמה שגיאות אפשריות החזירו את השגיאה שהוגדרה ראשונה בקובץ. אם מתרחשת שגיאה שאינה ברשימה, יש להחזיר MAP_ERROR.
- אין הגבלה על מספר האיברים במילון.
- במקרה של שגיאה יש לשמור על שלמות מבנה הנתונים ולוודא שאין דליפות זיכרון.
- במידה ו-mapPut או mapRemove מקבלות NULL כמפתח ו/או data, החזירו MAP_NULL_ARGUMENT.
- בתיעוד המופיע ב-map.h עבור חלק מהפונקציות כתוב שהאיטרטור במצב לא מוגדר אחרי הקריאה לפונקציה, המשמעות היא שכאשר איטרטור נמצא במצב זה, אסור למשתמש להניח משהו לגביו, כלומר שאינכם צריכים להבטיח שום דבר בנוגע לערך האיטרטור ואתם יכולים לשנות אותו כרצונכם.
- אם המשתמש קורא ל-mapPut עם מפתח שכבר קיים, המידע שקיים אמור להיות מוחלף בערך החדש והפונקציה תחזיר MAP_SUCCESS.
- הפונקציה mapGet מחזירה את הערך עצמו (ולא עותק).