

מבני נתונים

טבלאות מסכמות

ניר אדר

מסמך זה הורד מהאתר <http://underwar.livedns.co.il> אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר. מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מרב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

כל הזכויות שמורות לניר אדר

Nir Adar

Email: underwar@hotmail.com

Home Page: <http://underwar.livedns.co.il>

סיכום סיבוכיות מבני נתונים

מחסנית

פעולה	מימוש בעזרת מערך	מימוש בעזרת רשימה מקושרת חד כיוונית
create(S)	זמן $O(1)$, מקום $O(N)$	זמן $O(1)$, מקום $O(1)$
push(S, x)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
top(S)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
pop(S)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
is-empty(S)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$

סה"כ זיכרון: $O(N)$ עבור מערך, $O(n)$ עבור רשימה מקושרת חד כיוונית.

תור

פעולה	מימוש בעזרת מערך	מימוש בעזרת רשימה מקושרת חד כיוונית
create(Q)	זמן $O(1)$, מקום $O(N)$	זמן $O(1)$, מקום $O(1)$
head(Q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
enqueue(Q, x)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
dequeue(Q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
is-empty(Q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$

סה"כ זיכרון: $O(N)$ עבור מערך, $O(n)$ עבור רשימה מקושרת חד כיוונית.

דו תור

פעולה	מימוש בעזרת מערך	מימוש בעזרת רשימה מקושרת דו כיוונית
create(Q)	זמן $O(1)$, מקום $O(N)$	זמן $O(1)$, מקום $O(1)$
insert_head(Q, x)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
insert_tail(Q, x)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
delete_head(Q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
delete_tail(Q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
read_head(Q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
read_tail(Q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
is-empty(Q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$

סה"כ זיכרון: $O(N)$ עבור מערך, $O(n)$ עבור רשימה מקושרת דו כיוונית.

רשימה מקושרת

סיבוכיות	פעולה
זמן $O(1)$, מקום $O(1)$	init()
זמן $O(n)$, מקום $O(1)$	find(x, head)
זמן $O(1)$, מקום $O(1)$	delete(t)
זמן $O(1)$, מקום $O(1)$	insert(t, x)

סה"כ זיכרון: $O(n)$

עץ חיפוש בינארי

סיבוכיות	פעולה
זמן $O(1)$, מקום $O(1)$	init()
זמן $O(\log n)$ בממוצע, מקום $O(\log n)$	find(x, head)
זמן $O(\log n)$ בממוצע, מקום $O(\log n)$	delete(x)
זמן $O(\log n)$ בממוצע, מקום $O(\log n)$	insert(x)

סה"כ זיכרון: $O(n)$

עץ AVL

סיבוכיות	פעולה
זמן $O(1)$, מקום $O(1)$	init()
זמן $O(\log n)$, מקום $O(\log n)$	find(x, head)
זמן $O(\log n)$, מקום $O(\log n)$	delete(x)
זמן $O(\log n)$, מקום $O(\log n)$	insert(x)

סה"כ זיכרון: $O(n)$

עץ 2-3

סיבוכיות	פעולה
זמן $O(1)$, מקום $O(1)$	init()
זמן $O(\log n)$, מקום $O(\log n)$	find(x, head)
זמן $O(\log n)$, מקום $O(\log n)$	delete(x)
זמן $O(\log n)$, מקום $O(\log n)$	insert(x)

סה"כ זיכרון: $O(n)$

רשימת דילוגים רנדומאלית

פעולה	ללא מצביעים כלפי מעלה	עם מצביעים כלפי מעלה
init()	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
find(x, head)	זמן $O(\log n)$ בממוצע, מקום $O(1)$	זמן $O(\log n)$ בממוצע, מקום $O(1)$
delete(x)	זמן $O(\log n)$ בממוצע, מקום $O(1)$	זמן $O(\log n)$ בממוצע, מקום $O(1)$
insert(x)	זמן $O(\log n)$ בממוצע, מקום $O(\log n)$	זמן $O(\log n)$ בממוצע, מקום $O(1)$

סה"כ זיכרון: $O(n)$

רשימת דילוגים דטרמיניסטית

פעולה	ללא מצביעים כלפי מעלה	עם מצביעים כלפי מעלה
init()	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
find(x, head)	זמן $O(\log n)$, מקום $O(1)$	זמן $O(\log n)$, מקום $O(1)$
delete(x)	זמן $O(\log n)$, מקום $O(\log n)$	זמן $O(\log n)$, מקום $O(1)$
insert(x)	זמן $O(\log n)$, מקום $O(\log n)$	זמן $O(\log n)$, מקום $O(1)$

סה"כ זיכרון: $O(n)$

טבלת ערבול (Hash-Table)

פעולה	סיבוכיות
init()	זמן $O(N)$, מקום $O(1)$
find(x, head)	זמן $O(1)$ בממוצע, $O(n)$ במקרה הגרוע, מקום $O(1)$
delete(x)	זמן $O(1)$ בממוצע, $O(n)$ במקרה הגרוע, מקום $O(1)$
insert(x)	זמן $O(1)$, מקום $O(1)$

סה"כ זיכרון: $O(N+n)$

- על ידי שימוש בעץ מאוזן בכל כניסה ניתן לחסום את הזמן הגרוע על ידי $O(\log n)$.

תור עדיפויות / ערימה

פעולה	מימוש בעזרת עץ מאוזן	מימוש בעזרת עץ כמעט שלם
make_heap(Q)	זמן $O(n \log n)$, מקום $O(n)$	זמן $O(n)$, מקום $O(\log n)$
insert(x, Q)	זמן $O(\log n)$, מקום $O(\log n)$	זמן $O(\log n)$, מקום $O(1)$
max(Q)	זמן $O(\log n)$, מקום $O(\log n)$	זמן $O(1)$, מקום $O(1)$
del_max(Q)	זמן $O(\log n)$, מקום $O(\log n)$	זמן $O(\log n)$, מקום $O(\log n)$

סה"כ זיכרון: $O(n)$

קבוצות זרות - Union Find

פעולה	מערך בו עבור כל איבר נשמרת הקבוצה שלו	כל קבוצה ממומשת כרשימה מעגלית, רשימה מקושרת של קבוצות
MakeSet(i)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
Find(i)	זמן $O(1)$, מקום $O(1)$	זמן $O(n)$, מקום $O(1)$
Union(p, q)	זמן $O(N)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$

סה"כ זיכרון: עבור מערך: $O(N)$, עבור רשימה מקושרת מעגלית: $O(n)$.

פעולה	כל קבוצה כרשימה מקושרת, מערך המכיל מצביעים אל כל האיברים	כל קבוצה כרשימה מקושרת, מערך המכיל מצביעים אל כל האיברים, ממזגים קבוצות קטנות אל גדולות כל פעם
MakeSet(i)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
Find(i)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
Union(p, q)	זמן $O(n)$, מקום $O(1)$	זמן $O(\log n)$ משוערך, מקום $O(1)$

סה"כ זיכרון עבור מימושים אלו: $O(N)$

פעולה	כל קבוצה כעץ הפוך, מערך המכיל מצביעים אל כל האיברים	כל קבוצה כעץ הפוך, מערך המכיל מצביעים אל כל האיברים, מאחדים כל פעם קבוצות קטנות אל גדולות
MakeSet(i)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$
Find(i)	זמן $O(h)$, מקום $O(1)$	זמן $O(\log n)$, מקום $O(1)$
Union(p, q)	זמן $O(1)$, מקום $O(1)$	זמן $O(1)$, מקום $O(1)$

סה"כ זיכרון עבור מימושים אלו: $O(N)$

פעולה	כל קבוצה כעץ הפוך, מערך המכיל מצביעים אל כל האיברים, מאחדים כל פעם קבוצות קטנות אל גדולות, עם כיווץ מסלולים
MakeSet(i)	זמן $O(\log^* n)$ משוערך, מקום $O(1)$
Find(i)	זמן $O(\log^* n)$ משוערך, מקום $O(1)$
Union(p, q)	זמן $O(\log^* n)$ משוערך, מקום $O(1)$

סה"כ זיכרון עבור מימוש זה: $O(N)$

- ניתן להשתמש ב-Union-Find רק כאשר מספר הקבוצות המקסימאלי קבוע מראש.
- המימושים מניחים כי זמן הגישה אל קבוצה בהינתן שמה הינו $O(1)$.

אלגוריתמי מיון

שם המיון	זמן	מקום נוסף	הערות
Bubble Sort	$O(n^2)$	$O(1)$	מתבסס על החלפת איברים סמוכים על בסיס השוואות. בכל איטרציה מובא האיבר הגדול ביותר לקצה המערך, והמערך שמתופל באיטרציה הבאה "קטן" ב-1.
Heap Sort	$O(n \log n)$	$O(1)$	מיון המתבסס על מבנה הנתונים ערימה. בתחילה מאוחסנים הנתונים במבנה ערימה בזמן $O(n)$. לאחר מכן בכל איטרציה מוצא האיבר המקסימאלי ומתבצעת פעולת עדכון לשמירת חוקיות הערימה ($O(\log n)$). סה"כ מוצאים n איברים, ולכן הסיבוכיות היא $O(n \log n)$.
Quick Sort רנדומאלי	$O(n \log n)$ בממוצע, $O(n^2)$ במקרה הגרוע	$O(n)$ במימוש הרגיל, $O(\log n)$ אם מעלימים את רקורסיית הזנב	בכל איטרציה נבחר איבר ציר באופן אקראי והאיברים במערך מוסדרים כך שמשמאל לאיבר הציר כל האיברים הקטנים ממנו, ומימין האיברים הגדולים או שווים לו. לאחר מכן נשלחים שני המערכים שנוצרו בקריאה רקורסיבית לתהליך דומה.
Quick Sort דטרמיניסטי	$O(n \log n)$	$O(\log n)$	בכל איטרציה נבחר החציון בצורה דטרמיניסטית, ולכן מושג חסם עליון טוב יותר על זמן המיון.
Merge Sort	$O(n \log n)$	$O(n)$	פונקצית merge - ממצגת שני מערכים ממוינים (שאורכם יחד k) בזמן $O(k)$. תהליך המיון כולל חלוקה של מערך הקלט לתתי מערכים, עד שאורך כל מערך הוא 1, ולאחר מכן מיזוג של כל שני מערכים. באיטרציה הבאה ימוזגו ע"י merge המערכים בגודל 2 וכך הלאה.
Bucket Sort	$O(n + k)$	$O(1)$	מיון היעיל עבור מיון n מספרים שונים בתחום $1..k$. מיון זה הינו מיון יציב.
Radix Sort	$O(n)$ אם הבסיס קבוע, אחרת $O(d(n + b))$ כאשר d הוא מספר הספרות - n הוא הבסיס	$O(1)$	מיון זה הינו מיון יציב.