חלק ראשון

Add		
n	רשימה מקושרת	עץ חיפוש מאוזן
50	9	48
150	5	54
250	5	53
350	6	60
450	6	54
550	8	78
650	5	60
750	5	62
850	5	62
950	5	59
1050	6	79
1150	6	64
1250	6	70
1350	5	66
1450	6	73
1550	6	68
1650	5	66
1750	7	170
1850	6	64
1950	5	68
2050	5	67
2150	5	81
2250	6	73
2350	8	74
2450	5	74
2550	10	74
2650	5	75
2750	6	70
2850	7	76
2950	9	88
3050	6	75
3150	5	79
3250	5	71
3350	5	75
3450	7	72
3550	5	75
3650	6	71
3750	6	77
3850	5	74
3950	5	77
4050	8	74
4150	5	76
4250	5	84
4350	6	80

84	5	4450
77	5	4550
88	8	4650
81	5	4750
82	5	4850
85	6	4950
80	5	5050
80	5	5150
79	11	5250
95	5	5350
78	5	5450
80	5	5550
86	7	5650
81	5	5750
128	7	5850
79	5	5950
83	5	6050
76	5	6150
79	5	6250
94	7	6350
74	6	6450
78	6	6550
79	6	6650
84	6	6750
79	5	6850
83	7	6950
79	5	7050
90	5	7150
84	14	7250
80	6	7350
85	5	7450
90	7	7550
87	6	7650
85	6	7750
81	9	7850
87	5	7950
79	6	8050
81	7	8150
89	5	8250
91	6	8350
93	6	8450
90	5	8550
88	5	8650
97	7	8750
92	6	8850
99	5	8950
93	6	9050
88	5	9150
93	5	9250

89	7	9350
91	5	9450
89	6	9550
101	6	9650
95	6	9750
91	7	9850
99	5	9950

Delete		
עץ חיפוש מאוזן	רשימה מקושרת	n
56	3886	9950
50	23	9850
44	22719	9750
48	25	9650
45	10592	9550
50	23	9450
43	3836	9350
46	24	9250
43	17748	9150
53	24	9050
48	3703	8950
47	24	8850
40	17478	8750
49	24	8650
43	3454	8550
49	24	8450
43	17394	8350
47	29	8250
44	3187	8150
44	23	8050
43	17128	7950
47	23	7850
39	3060	7750
44	24	7650
41	2990	7550
46	24	7450
37	18961	7350
47	24	7250
39	2942	7150
46	24	7050
51	16787	6950
51	24	6850
45	2713	6750
48	24	6650
41	12620	6550
50	24	6450
42	2485	6350

46	23	6250
43	2409	6150
45	24	6050
42	2400	5950
44	24	5850
38	11005	5750
41	25	5650
39	2261	5550
46	24	5450
38	2157	5350
44	23	5250
38	2121	5150
42	25	5050
39	1997	4950
50	24	4850
36	15880	4750
43	24	4650
40	1801	4550
44	23	4450
38	1713	4350
47	23	4250
42	15951	4150
40	23	4050
41	1587	3950
40	23	3850
39	1497	3750
38	24	3650
38	1404	3550
39	24	3450
39	1352	3350
41	24	3250
38	15265	3150
38	24	3050
39	1165	2950
36	23	2850
36	1081	2750
39	23	2650
35	1006	2550
47	23	2450
37	924	2350
41	23	2250
42	847	2150
37	14036	2050
39	810	1950
37	24	1850
37	730	1750
36	24	1650
34	647	1550
36	24	1450

34	556	1350
37	24	1250
34	475	1150
34	23	1050
34	375	950
33	24	850
34	306	750
33	23	650
32	213	550
33	23	450
30	137	350
32	23	250
27	61	150
21	23	50

Find		
עץ חיפוש מאוזן	רשימה מקושרת	n
8	23	100
8	22	100
8	59	200
8	22	200
8	108	300
9	23	300
9	142	400
9	23	400
8	176	500
8	21	500
8	219	600
9	23	600
8	266	700
9	22	700
12	294	800
9	21	800
9	332	900
8	22	900
8	375	1000
8	21	1000
10	410	1100
9	21	1100
8	456	1200
12	22	1200
8	496	1300
9	22	1300
7	531	1400
9	22	1400
7	565	1500
9	21	1500

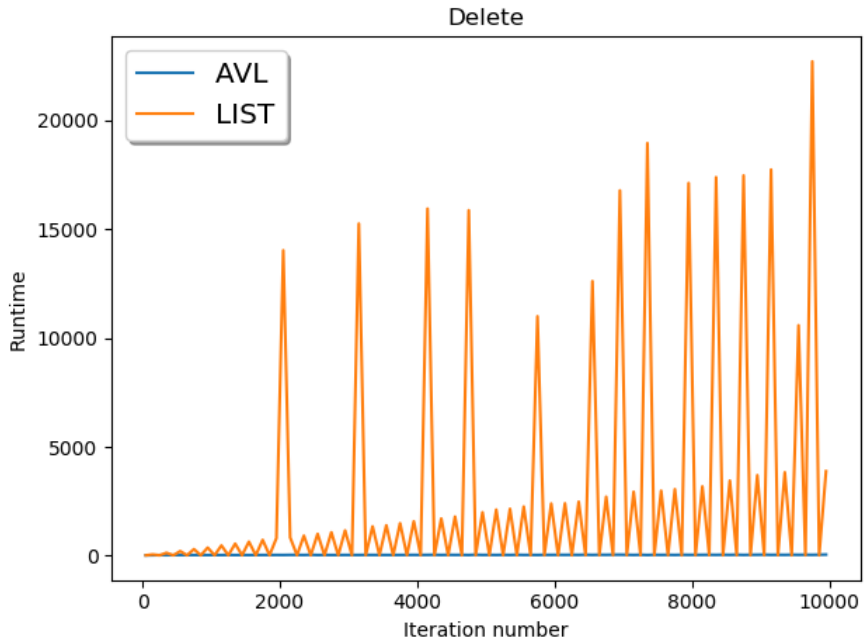
10	607	1600
9	22	1600
10	646	1700
9	21	1700
9	684	1800
9	21	1800
11	724	1900
10	21	1900
9	763	2000
9	21	2000
9	10804	2100
11	22	2100
9	843	2200
10	21	2200
9	879	2300
10	21	2300
8	917	2400
10	21	2400
8	955	2500
10	22	2500
11	999	2600
10	22	2600
9	1051	2700
10	22	2700
9	1074	2800
10	22	2800
9	1116	2900
12	21	2900
9	1225	3000
10	33	3000
11	11255	3100
9	22	3100
14	1280	3200
11	22	3200
13	1267	3300
14	21	3300
10	1311	3400
10	21	3400
9	1360	3500
10	22	3500
9	1396	3600
9	21	3600
9	1426	3700
9	22	3700
9	3498	3800
10	23	3800
9	1601	3900
10	21	3900
9	1541	4000

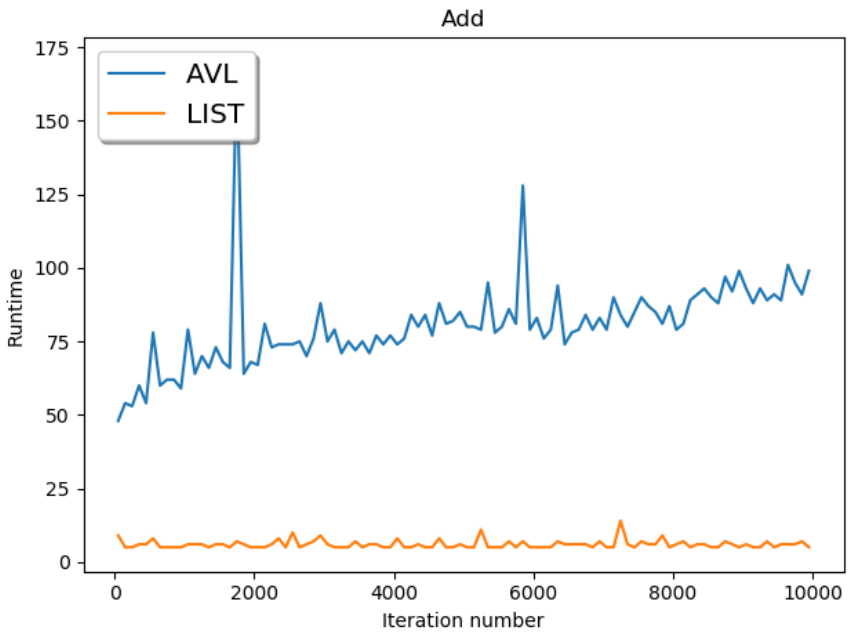
10	21	4000
25	1635	4100
10	22	4100
12	1663	4200
15	22	4200
10	1688	4300
11	23	4300
9	1752	4400
11	22	4400
12	21750	4500
10	21	4500
12	1786	4600
15	22	4600
10	1828	4700
11	22	4700
10	1966	4800
14	23	4800
12	1967	4900
14	22	4900
12	1961	5000
13	22	5000
13	2067	5100
15	22	5100
12	2087	5200
14	23	5200
12	2237	5300
14	22	5300
12	2191	5400
16	10030	5400
12	2133	5500
14	22	5500
12	2236	5600
14	22	5600
10	2311	5700
11	21	5700
11	2472	5800
13	23	5800
10	12339	5900
13	22	5900
13	2371	6000
13	22	6000
11	2465	6100
11	33	6100
12	2422	6200
11	22	6200
12	12508	6300
10	22	6300
13	2629	6400
11	22	6400

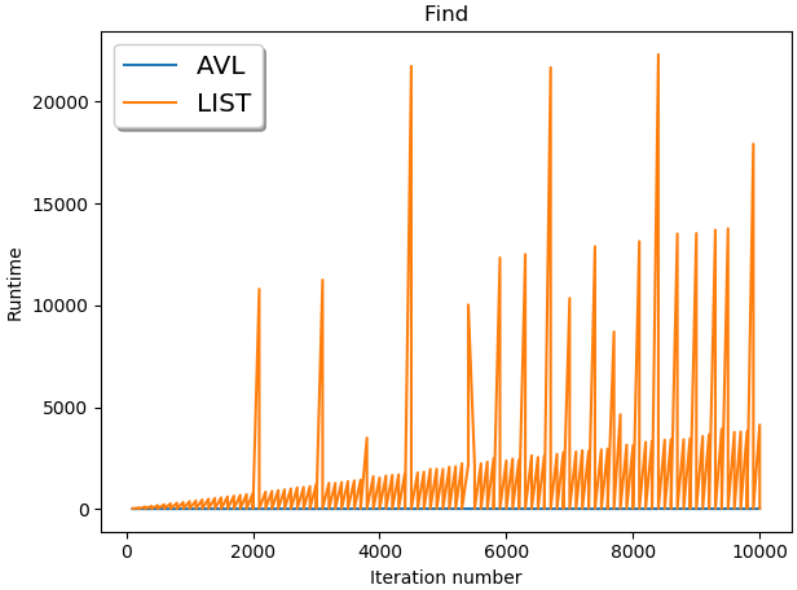
11	2543	6500
11	22	6500
10	2618	6600
10	24	6600
10	21683	6700
11	21	6700
10	2698	6800
11	23	6800
9	2777	6900
11	22	6900
10	10355	7000
10	22	7000
10	2810	7100
10	22	7100
10	2875	7200
11	21	7200
9	2844	7300
10	22	7300
9	12895	7400
10	21	7400
9	2920	7500
11	21	7500
10	2960	7600
12	22	7600
10	8710	7700
10	22	7700
10	4642	7800
10	26	7800
9	3156	7900
10	22	7900
9	3118	8000
11	22	8000
9	13147	8100
10	22	8100
9	3288	8200
11	22	8200
10	3325	8300
16	33	8300
10	22325	8400
23	21	8400
10	3389	8500
16	21	8500
10	3423	8600
14	22	8600
10	13520	8700
14	21	8700
10	3418	8800
15	21	8800
10	3461	8900

14	22	8900
10	13541	9000
15	22	9000
9	3575	9100
13	22	9100
10	3656	9200
15	22	9200
10	13704	9300
14	22	9300
9	3933	9400
15	23	9400
10	13774	9500
13	22	9500
9	3775	9600
12	22	9600
9	3794	9700
15	22	9700
10	3804	9800
13	21	9800
10	17928	9900
15	22	9900
10	4135	10000
13	25	10000

להלן הגרפים:


Delete
AVL
LIST
Runtime
Iteration number
0
5000
10000
15000
20000
0
2000
4000
6000
8000
10000


Add
AVL
LIST
Runtime
Iteration number
0
25
50
75
100
125
150
175
0
2000
4000
6000
8000
10000


Find
AVL
LIST
Runtime
Iteration number
0
5000
10000
15000
20000
0
2000
4000
6000
8000
10000

חלק שני:

מבנה כללי:

מבנה מאגר התמונות שומר **עץ** AVL (מעתה ייקרא "$Images$") בעל מפתחות מספריים המסמלים את "מזהה התמונה" וערכים מסוג "תמונה".

כל ערך תמונה מכיל בתוכו מזהה תמונה, וכן **מערך ורשימה מקושרת דו כיוונית**.

המערך (מעתה ייקרא "$segments$") נועד לסמל את הסגמנטים של התמונה ואת התיוגים שלהם.

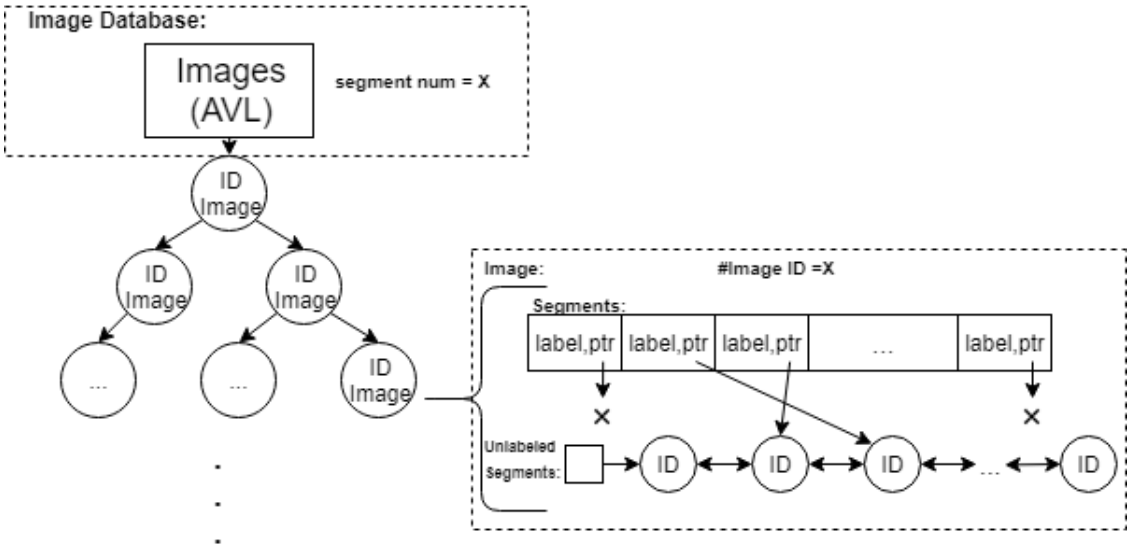
הרשימה המקושרת (מעתה תיקרא "$unlabeled$") נועדה להכיל מידע על הסגמנטים חסרי התיוג בתמונה.

המערך באורך זהה למספר הסגמנטים, כאשר כל תא במערך תואם באינדקס שלו למספר הסגמנט שהוא מייצג.

עבור כל תא במערך, הסגמנט שהוא מייצג יכול להיות משני סוגים:

- <u>עבור סגמנט מתויג:</u> התא המתאים במערך יכיל את הערך המספרי של התיוג ומצביע ל-$NULL$.
- <u>עבור סגמנט חסר תיוג:</u> התא המתאים במערך יכיל מצביע לחוליה מתאימה לסגמנט שתימצא ברשימה המקושרת, והערך המספרי בתא יהיה "ערך זבל" שאין להתייחס אליו כתיוג הסגמנט אם קיים בתא מצביע חוקי.

להלן תרשים הממחיש את המבנה הכללי:



מימוש הפעולות וסיבוכיות זמן:

$Init$:

מתבצע אתחול של העץ $Images$ כעץ ריק.
לאחר מכן שומרים בתוך המבנה את מספר הסגמנטים הקבוע לכל תמונה לשימוש עתידי.
מכיוון שאתחול עץ AVL והעתקת ערך מתבצעים ב־$O(1)$, זוהי גם הסיבוכיות הכוללת.

$AddImage$:

ראשית מתבצע חיפוש של התמונה בעץ. אם התמונה קיימת, מחזירים שגיאה.
אם התמונה שרוצים להוסיף היא אכן חדשה, מתבצעת יצירה של אובייקט התמונה שיש להכניס למבנה:
שומרים בתמונה את המזהה המספרי שלה, ולאחר מכן משום שביצירת תמונה, כל הסגמנטים בה חסרי תיוג, מכניסים בלולאה חוליה לרשימת $Unlabeled$ עבור כל סגמנט, המכילה את המזהה שלו.
במקביל, המצביע לכל חוליה נשמר בתא המתאים במערך $segments$.
מכיוון שהכנסת חוליה לרשימה מקושרת ושמירת במצביע שלה במערך המאפשר גישה ישירה מתבצעים ב־$O(1)$, נקבל שאתחול אובייקט תמונה מתבצע בסיבוכיות $O(n)$ כאשר עבור כל אחד מ־n הסגמנטים מתבצעות הפעולות הללו פעם אחת.
לאחר מכן מצביע לאובייקט התמונה החדש מוכנס לעץ $Images$ עם מזהה התמונה כמפתח.
סיבוכיות חיפוש והכנסה לעץ AVL כאשר במערכת k תמונות היא $O(\log(k))$.
לכן הסיבוכיות הכוללת היא: $O(\log(k)+n)$.

$DeleteImage$:

ראשית מתבצע חיפוש התמונה שיש למחוק בעץ $Images$ לפי המזהה שלה.
סיבוכיות החיפוש בעץ AVL בעל k תמונות: $O(\log(k))$.
אם התמונה נמצאה, מוציאים אותה מהעץ (בסיבוכיות $O(\log(k))$ בעץ AVL) ומוחקים את אובייקט ה־$Image$.
במחיקת האובייקט, מתבצע שחרור בלולאה של כל סגמנט במערך, ולאחר מכן שחרור כל חוליה ברשימה המקושרת.
מכיוון ששניהם מכילים לכל היותר n סגמנטים, ושחרור תא במערך או חוליה ברשימה מתבצע ב־$O(1)$, סיבוכיות ההריסה היא $O(n)$.
לכן הסיבוכיות הכוללת היא: $O(\log(k)+n)$.

$AddLabel$:

ראשית מתבצע חיפוש בעץ $Images$ של התמונה שלתוכה יש להוסיף תיוג.

חיפוש בעץ AVL בעל k תמונות הוא בסיבוכיות $O(\log(k))$.

אם התמונה נמצאה, מתבצעת גישה ישירה לתא במערך $segments$ שלה שמייצג את הסגמנט שיש לתייג.

אם מהבדיקה עולה שהסגמנט כבר מתויג (המצביע בתא $NULL$) מוחזרת שגיאה.

אם הסגמנט אינו מתויג, מתייגים את התא המתאים במערך ובאמצעות המצביע השמור בתא לחוליה המתאימה ברשימת $Unlabeled$ ניתן להוציא ולהרוס אותה בסיבוכיות $O(1)$ מכיוון שזו רשימה דו כיוונית.

לאחר מכן מסמנים את הסגמנט בתא המתאים כ"מתויג" על ידי השמה של $NULL$ למצביע החוליה בתוכו.

מכיוון שלאחר מציאת התמונה, כל הפעולות מתבצעות בגישה ישירה ובסיבוכיות $O(1)$, הסיבוכיות הכוללת היא: $O(\log(k))$ (ובפרט $O(\log(k)+n)$)

$GetLabel$:

ראשית מתבצע חיפוש בעץ $Images$ של התמונה שממנה רוצים לקרוא את התיוג.

חיפוש בעץ AVL בעל k תמונות הוא בסיבוכיות $O(\log(k))$.

אם התמונה לא נמצאה, מוחזרת שגיאה, ואחרת מבצעים גישה ישירה בתמונה למערך הסגמנטים ובודקים את מצב הסגמנט המבוקש.

אם הסגמנט המבוקש אינו מתויג, נדע זאת בסיבוכיות $O(1)$ מכך שהמצביע בתא המתאים במערך לא יהיה $NULL$, ונחזיר שגיאה.

אם הסגמנט המבוקש מתויג, נקרא את התיוג בגישה ישרה מהמערך ב־$O(1)$ ונחזיר אותו.

בסך הכול חיפוש ומספר קבוע של גישות ישירות הם בסיבוכיות כוללת של $O(\log(k))$.

$DeleteLabel$:

ראשית מתבצע חיפוש בעץ $Images$ של התמונה שבה מבקשים למחוק התיוג.

חיפוש בעץ AVL בעל k תמונות הוא בסיבוכיות $O(\log(k))$.

אם התמונה לא נמצאה מוחזרת שגיאה, ואחרת מבצעים גישה ישירה בתמונה למערך הסגמנטים ובודקים את מצב הסגמנט המבוקש.

בדומה לפעולה ב־$GetLabel$, אם הסגמנט אינו מתויג נוכל להחזיר שגיאה בסיבוכיות $O(1)$.

אם הסגמנט מתויג, כל שיש לעשות כדי לסמנו כ"לא מתויג" הוא ליצור חוליה חדשה ל־$Unlabeled$ עם מספר הסגמנט, להכניס אותה לרשימה בסיבוכיות $O(1)$ (כי זו הסיבוכיות שבה נתבקשנו לממש הכנסה לרשימה), ולאחר מכן להכניס את המצביע אליה לתא במערך $segments$ המסמל את הסגמנט שאת תיוגו מחקנו. מעתה בכל גישה למבנה הנתונים נוכל לראות שמצביע זה אינו $NULL$ ולכן לפי הצורה בה הגדרנו את המבנה, הסגמנט אכן אינו מתויג.

מספר קבוע של פעולות השמה והכנסת חוליה לרשימה מקושרת מתבצעים ב־$O(1)$.

לכן הסיבוכיות הכוללת היא $O(\log(k))$.

$GetAllUnlabeledSegments$:

ראשית מתבצע חיפוש בעץ $Images$ של התמונה שממנה יש למצוא סגמנטים.
חיפוש בעץ AVL בעל k תמונות הוא בסיבוכיות $O(\log(k))$.
אם התמונה אינה קיימת, נחזיר שגיאה ונבצע השמה של ערכי $NULL$ מתאימים ב־$O(1)$ למצביע שהועבר לפונקציה.
לאחר מכן נבצע ספירה של האזורים שאינם מתויגים בתמונה שנמצאה.
הספירה מתבצעת בסיבוכיות $O(1)$ משום שהרשימה המקושרת $Unlabeled$ ממומשת בחלק 1 באופן ששומר את שדה הגודל שלה ומחזיר אותו ב־$O(1)$.
לכן נוכל לדעת ב־$O(1)$ האם לא קיים בתמונה אזור פנוי (גודל $Unlabeled$ יהיה 0).
אם קיימים בתמונה אזורים פנויים, נקצה מקום למערך שיכיל את מספריהם (שאת אורכו השגנו לאחר הספירה).
נבצע סריקה של הרשימה $Unlabeled$ ונעתיק את תוכנה למערך (כלומר מקבלים מערך של כל מספרי הסגמנטים הלא מתויגים).
ההעתקה מתבצעת ב־$O(1)$ מכיוון שהערכים ב־$Unlabeled$ הם רק int.
לבסוף נחזיר את המערך החדש שבנינו שמכיל את כל הסגמנטים הלא מתויגים, דרך המצביע שהועבר לפונקציה.
לפי הגדרת הרשימה $Unlabeled$ כרשימה שמכילה רק את מספרי הסגמנטים הלא מתויגים: נקבל שאורכה s, ומכיוון שעבור כל חוליה שלה מתבצעות פעולות ב־$O(1)$ נקבל שסיבוכיות המעבר על הרשימה היא $O(s)$.
לכן הסיבוכיות הכוללת היא $O(\log(k)+s)$.

$GetAllSegmentsByLabel$:

ראשית נבצע סיור על העץ $Images$ על מנת לחלץ ממנו מערך המכיל מצביעים לכל התמונות.
סיור מתבצע על עץ חיפוש בעל k תמונות ב־$O(k)$.
לאחר מכן נעבור על מערך התמונות בלולאה ועבור כל תמונה נסכום את מספר הסגמנטים שהם בעלי התיוג הרצוי.
עבור כל תמונה, הספירה מתבצעת על ידי מעבר בלולאה על מערך $segments$ שלה והחזקת מונה של מספר הסגמנטים שנמצאו מתויגים ובעלי התיוג הרצוי.
לכן עבור כל תמונה נוכל לספור את הסגמנטים הרלוונטיים בה ב־$O(n)$.
לכן הסכימה על כל k התמונות מתבצעת ב־$O(k \cdot n)$.
לאחר מכן נקצה שני מערכי מספרים בגודל שחילצנו מהסכימה (הקצאה ב־$O(1)$).
נבצע שוב איטרציה על מערך התמונות. עבור כל תמונה, נחלץ מתוכה מערך של כל הסגמנטים שהם בעלי התיוג הרצוי.
הדבר מתבצע על ידי איטרציה על מערך $segments$ והעתקת מספרי הסגמנטים הרצויים למערך נפרד.
לאחר בניית מערך הסגמנטים בעלי התיוג הרצוי, נבצע איטרציה עליו, ועבור כל סגמנט עם התיוג הרצוי, נעתיק את מספרו ואת מספר התמונה שבה הוא נמצא למערכים שבנינו.
על מנת לחלץ את מערך הסגמנטים הרצויים בכל תמונה, הסיבוכיות היא סיבוכיות איטרציה על מערך $segments$ וביצוע העתקות ב־$O(1)$ עבור כל תא בו, כלומר סיבוכיות $O(n)$.
לאחר מכן סיבוכיות איטרציה נוספת על מערך הסגמנטים הרצויים גם היא ב־$O(n)$ משום שמערך זה מכיל לכל היותר את כל הסגמנטים בתמונה.

לכן עבור כל תמונה מצבעים פעולות בסיבוכיות $O(n)$, כלומר הסיבוכיות הכוללת עבור כל התמונות היא $O(k \cdot n)$ מכיוון שעבור כל סגמנט מבצעים מספר קבוע של פעולות ובדיקות ב־$O(1)$.

מכיוון שבעץ $Images$, התמונות שמורות ממוינות לפי המזהים שלהם מתכונות עץ חיפוש, וסיור $Inorder$ שומר על תכונה זו כאשר אנו מעתיקים את התמונות למערך, וכמו כן בתוך כל תמונה אנו מבצעים איטרציה על מערך ה־$segments$ מההתחלה לסוף, נקבל שמערכי התוצאה ממוינים כמבוקש לפי מזהה תמונה בסדר עולה, ועם סידור פנימי בסדר עולה של סגמנטים בתמונה משותפת.

לבסוף נהרוס את מערך מצביעי התמונות שחילצנו מהעץ איבר־איבר. מכיוון שהמערך מכיל מצביעים לתמונות, ולא את התמונות עצמן, נוכל להרוס כל מצביע (תא במערך) בסיבוכיות $O(1)$. כלומר המערך כולו נהרס בסיבוכיות $O(k)$: אורך המערך.

בסך הכול ביצענו רצף פעולות בסיבוכיות $O(k + k \cdot n + k \cdot n + k) = O(k \cdot n)$.

$Quit$:

מתבצע מעבר על העץ $Images$ ושחרור כל תמונה בתוכו.

במימוש מחלק 1 של העץ, סיבוכיות השחרור של מבנה העץ בעל k התמונות היא $O(k)$.

אך בנוסף עבור כל תמונה בעץ קיימת סיבוכיות של שחרור השדות בתמונה:

עבור כל תמונה בעץ נעבור על מערך הסגמנטים בלולאה ונשחרר את הזיכרון המוקצה לכל תא ב־$O(1)$.

כלומר סיבוכיות ההריסה של המערך באורך n היא $O(n)$.

לאחר מכן נעבור חוליה־חוליה על הרשימה המקושרת $Unlabeled$ ונהרוס כל חוליה בסיבוכיות $O(1)$.

הרשימה מכילה לכל היותר את כל n הסגמנטים בתמונה, לכן הסיבוכיות של הריסתה היא $O(n)$.

בסך הכול סיבוכיות הריסה של כל תמונה היא $O(n)$, ולכן סיבוכיות הריסת כל k התמונות היא $O(k \cdot n)$, וכזו גם הסיבוכיות הכוללת.

סיבוכיות מקום:

במבנה בעל k תמונות קיים עץ AVL ששומר אותן, ולכן סיבוכיות המקום שלו היא $O(k)$.

בנוסף, סיבוכיות המקום בכל תמונה היא סיבוכיות שמירה של מערך באורך n הסגמנטים, רשימה שמכילה לכל היותר את כל הסגמנטים ולכן באורך לכל היותר n, וכן מספר קבוע של ערכים ומצביעים.

לכן סיבוכיות המקום של אחסון תמונה היא $O(n)$, ולכן סיבוכיות אחסון כל k התמונות, שהיא הסיבוכיות הכוללת של המבנה, היא $O(k \cdot n)$.

כעת נותר להוכיח שסיבוכיות המקום של רקורסיות והקצאות מערכים (פרט למערכים השמורים במבנה) בפעולות אינן חורגות מסיבוכיות זו:

$Init$:

אין הקצאת מערכים או מקום נוסף, וללא רקורסיות.

$Quit, AddImage, DeleteImage, AddLabel, GetLabel, DeleteLabel$:

בכל הפונקציות הללו אין הקצאת מקום נוסף.

הרקורסיה שמתבצעת בהן היא רקורסיה בפעולות הריסה, חיפוש, הכנסה והוצאה לעץ AVL.

מכיוון שהעץ בעומק k, מתכונות עץ AVL הפעולות הללו הן בעומק לכל היותר גובה העץ, כלומר $\log(k)$.

לכן סיבוכיות המקום הנוסף ברקורסיה בכל הפעולות הללו היא $O(\log(k))$ ובפרט $O(k \cdot n)$.

$GetAllUnlabeledSegments$:

כמו בפעולות הקודמות, ברקורסיה מבצעים פעולות על עץ AVL בסיבוכיות מקום $O(\log(k))$.

בנוסף מתבצעת הקצאה של מערך הסגמנטים הלא מתויגים, שהוא לכל היותר באורך $k \cdot n$ (אם כל הסגמנטים בכל התמונות אינם מתויגים).

לכן סיבוכיות המקום הנוסף היא $O(k \cdot n)$.

$GetAllSegmentsByLabel$:

באופן דומה לפעולה הקודמת, קיימת סיבוכיות מקום רקורסיבית של $O(\log(k))$.

בנוסף מקצים מערך זמני לשמירה של כל התמונות, באורך k ולכן בסיבוכיות מקום נוסף $O(k)$.

מערך זה נהרס בסוף הפעולה, ולכן הסיבוכיות אינה מצטברת במבנה.

בנוסף מתבצעת הקצאה של המערכים המוחזרים מהפונקציה, שהם לכל היותר מערך של כל k התמונות ומערך של כל $k \cdot n$ הסגמנטים.

כלומר סיבוכיות המקום הנוסף היא עדיין $O(k \cdot n)$.

בסך הכול הראינו שסיבוכיות המקום במבנה וסיבוכיות המקום הנוסף בפונקציות היא $O(k \cdot n)$, ולכן זו סיבוכיות המקום הכוללת כנדרש.