

יחידה 9 יעילות

9.1 הקדמה

מדדי סיבוכיות

סיבוכיות זמן - *time complexity*

צמן החישוב - מושך הזמן בו מתבצעת התכנית

סיבוכיות מקום - *space complexity*

מקום קצתון - אודל זכרון המחשב שיש להקצות לצורך ביצוע התכנית

9.2 שיפורים

כיצד נמדוד יעילות?

אנחנו בריחז אנדוס אט מספר הפעולות שאנחנו מקבל אליו גור קצתון מסומן ואז לא קאונר הקלס

חישוב מספר הפעולות הבסיסיות

פעולה בסיסית = פעולה שלא גורת תלס, כמו חסור, חיסור, כפל, גילוף... לא "גילוף" הסכמה ס קמחוסה S - ח סה גלוי באונר של S

ד פעולת - 3 הצהרה, 1 השעה ← `int count, n, sum = 0;`

קמך תולסזה, יש 3 פעולות: `for (count = 0; count < n; count++)`

פעולת קצתה גלוי, פעולת חוסה, פעולת 1 השעה: `sum = sum + count;`

`System.out.println("sum = " + sum);`

פעולת חוסה

$$4 + 1 + 3 \cdot n + 1 = 3n + 6$$

הי כולסנה אנקציה ח פעולות

סז הנ מספר הפעולות:

לוקי סקווינז הנן של האלסורגז הנה. זקו קיסי שהוסז פוקציה של ח - כש-ח הוא אונר הקלס

9.4 מיון מערך $O(n^2)$

2. ЛІКУНІТ 9.6

דוגמא 4 - מערך שמהווה סדרה "עולה יורדת" ומחזיר את החסי הסקסיילי.
 * (סדרה עלתה יורדת = כל איבר מסוים קטן מן האיבר הבא, ואיבר נוסף)
 הם חלשים וגדלים - (גדלים באופן שני איברים רבנים גדלים).

[illegible]

צאס היינו צריכים למצוא את המס העיני, היינו יכולים לשמש זאת כפסגה אחת בלבד -
לכונן עי. עשי הקצוות יורגן. נאמן, שהסיקוניות כיון היא קבועה - $O(1)$.

דומה 5 - ג'ן ב' ח קומות - החל מאי 15 קומה ישר כדד השמירה ששירות חתונה?
 הפסגות: מ 28 סגור - (ח)

בנוי צורה - אך המעבר יש לנו שני דיוורים. כצד (בדיוק)?
 נזכור שגם לפני חיסוסי סיבירי בהתחלה ואז למדנו סדרתי. (אחרי שהלכנו להוסון, שרר)
 אולי זה לא ממש. הסיקוביות של היא צדן (א). אפשר גם לפחד 25 שטח-אנדרה צדן (ס).
 ההנהלה שהלכנו כן. כל זמננו, יהיה לפחד כל מה הנצור הראשון, ואז עברו
 צדן 9. קצתו לפני צדן הניצור השני. (קצתו הראשון קצתו 20, 30
 ו-6. שרר - ענינו, קצתו הניצור השני בין 39 31-5).
 הסיקוביות של הפבריון הפנה (היא) סיבירי, ריבועי - quadratic - (an) ,
 כי כל הוסי (השני של כל (א) ו-9 קצתו הניצור השני, אז נספור כי זה קצתו).

השוואת סדרי אודל

* אוקטובר 1971 =
מחזור

מחירי	1000	100	10	סיבוכיות	
1	1	1	1	$\left. \begin{array}{l} \text{זמן אקט} \\ \text{כתיבת אקט} \\ (\text{סקינים}) \end{array} \right\}$	
10	7	3	$\log n$		
31	6	3	\sqrt{n}		
1000	100	10	n		
3000	300	30	$n \log n$		
1,000,000	10,000	100	n^2	$\left. \begin{array}{l} \text{זמן אקט} \\ \text{קריאת אקט} \\ \text{סקינים} \end{array} \right\}$	
302 ק"מ	31 ס"מ	1024	2^n		
314 ק"מ	101 ס"מ	36×10^6	n^n		

דוגמא 2 - תכנית לחישוב ההפרש הגדול ביותר בין שני מספרים במחר.
 במקום למצוא δ המזער δ^2 במידה ונקודת אפס הנטול האפשרי, אפשר
 למצוא בעזרת אחת על δ^2 ^{האחרת אפשר} - בעזרת δ^2 ^{האחרת אפשר}, ולמצוא את המס' הנמצא.
 ואת המס' המזער. ונחשבים ההפרש ביניהם.
 יתרון - $O(n^2)$ - $O(n)$ - אסימטרית בקלות אלגוריתם.

דוגמא 3 - תגית המוצאת האם ישנם שני חסרונים שסכומם שווה למס ...
 שמתקבל כחסרטר - אם כן מחזורית true ואם לא - false (המספר החזין)
 # 1 - תגית המוצאת האם ישנם שני חסרונים שסכומם שווה למס ...

#2 - סיבוכיות (Complexity): כמות הזיכרון או הזמן הנדרשים לביצוע משימה. לדוגמה, חישוב 2^n הוא סיבוכי יותר מחישוב n^2 .

for (int i=0; i<a.length; i++)
if (binarySearch(a, num-a[i]))
return true;

כיוון שהמחלקה n פחותה מקצה המידות $\log n$,
 והשורה $\log n$ פחותה מ-1, נקבעת הפונקציה $\log n$ כפונקציה
 # צורה 3-0 פונקציה $O(n)$:
 מול הנוכחיות והפונקציה $\log n$ פחותה מ-1, נקבעת הפונקציה $\log n$ כפונקציה
 1-0, 1-1, 1-2, 1-3, 1-4, 1-5, 1-6, 1-7, 1-8, 1-9, 1-10, 1-11, 1-12, 1-13, 1-14, 1-15, 1-16, 1-17, 1-18, 1-19, 1-20, 1-21, 1-22, 1-23, 1-24, 1-25, 1-26, 1-27, 1-28, 1-29, 1-30, 1-31, 1-32, 1-33, 1-34, 1-35, 1-36, 1-37, 1-38, 1-39, 1-40, 1-41, 1-42, 1-43, 1-44, 1-45, 1-46, 1-47, 1-48, 1-49, 1-50, 1-51, 1-52, 1-53, 1-54, 1-55, 1-56, 1-57, 1-58, 1-59, 1-60, 1-61, 1-62, 1-63, 1-64, 1-65, 1-66, 1-67, 1-68, 1-69, 1-70, 1-71, 1-72, 1-73, 1-74, 1-75, 1-76, 1-77, 1-78, 1-79, 1-80, 1-81, 1-82, 1-83, 1-84, 1-85, 1-86, 1-87, 1-88, 1-89, 1-90, 1-91, 1-92, 1-93, 1-94, 1-95, 1-96, 1-97, 1-98, 1-99, 1-100, 1-101, 1-102, 1-103, 1-104, 1-105, 1-106, 1-107, 1-108, 1-109, 1-110, 1-111, 1-112, 1-113, 1-114, 1-115, 1-116, 1-117, 1-118, 1-119, 1-120, 1-121, 1-122, 1-123, 1-124, 1-125, 1-126, 1-127, 1-128, 1-129, 1-130, 1-131, 1-132, 1-133, 1-134, 1-135, 1-136, 1-137, 1-138, 1-139, 1-140, 1-141, 1-142, 1-143, 1-144, 1-145, 1-146, 1-147, 1-148, 1-149, 1-150, 1-151, 1-152, 1-153, 1-154, 1-155, 1-156, 1-157, 1-158, 1-159, 1-160, 1-161, 1-162, 1-163, 1-164, 1-165, 1-166, 1-167, 1-168, 1-169, 1-170, 1-171, 1-172, 1-173, 1-174, 1-175, 1-176, 1-177, 1-178, 1-179, 1-180, 1-181, 1-182, 1-183, 1-184, 1-185, 1-186, 1-187, 1-188, 1-189, 1-190, 1-191, 1-192, 1-193, 1-194, 1-195, 1-196, 1-197, 1-198, 1-199, 1-200, 1-201, 1-202, 1-203, 1-204, 1-205, 1-206, 1-207, 1-208, 1-209, 1-210, 1-211, 1-212, 1-213, 1-214, 1-215, 1-216, 1-217, 1-218, 1-219, 1-220, 1-221, 1-222, 1-223, 1-224, 1-225, 1-226, 1-227, 1-228, 1-229, 1-230, 1-231, 1-232, 1-233, 1-234, 1-235, 1-236, 1-237, 1-238, 1-239, 1-240, 1-241, 1-242, 1-243, 1-244, 1-245, 1-246, 1-247, 1-248, 1-249, 1-250, 1-251, 1-252, 1-253, 1-254, 1-255, 1-256, 1-257, 1-258, 1-259, 1-260, 1-261, 1-262, 1-263, 1-264, 1-265, 1-266, 1-267, 1-268, 1-269, 1-270, 1-271, 1-272, 1-273, 1-274, 1-275, 1-276, 1-277, 1-278, 1-279, 1-280, 1-281, 1-282, 1-283, 1-284, 1-285, 1-286, 1-287, 1-288, 1-289, 1-290, 1-291, 1-292, 1-293, 1-294, 1-295, 1-296, 1-297, 1-298, 1-299, 1-300, 1-301, 1-302, 1-303, 1-304, 1-305, 1-306, 1-307, 1-308, 1-309, 1-310, 1-311, 1-312, 1-313, 1-314, 1-315, 1-316, 1-317, 1-318, 1-319, 1-320, 1-321, 1-322, 1-323, 1-324, 1-325, 1-326, 1-327, 1-328, 1-329, 1-330, 1-331, 1-332, 1-333, 1-334, 1-335, 1-336, 1-337, 1-338, 1-339, 1-340, 1-341, 1-342, 1-343, 1-344, 1-345, 1-346, 1-347, 1-348, 1-349, 1-350, 1-351, 1-352, 1-353, 1-354, 1-355, 1-356, 1-357, 1-358, 1-359, 1-360, 1-361, 1-362, 1-363, 1-364, 1-365, 1-366, 1-367, 1-368, 1-369, 1-370, 1-371, 1-372, 1-373, 1-374, 1-375, 1-376, 1-377, 1-378, 1-379, 1-380, 1-381, 1-382, 1-383, 1-384, 1-385, 1-386, 1-387, 1-388, 1-389, 1-390, 1-391, 1-392, 1-393, 1-394, 1-395, 1-396, 1-397, 1-398, 1-399, 1-400, 1-401, 1-402, 1-403, 1-404, 1-405, 1-406, 1-407, 1-408, 1-409, 1-410, 1-411, 1-412, 1-413, 1-414, 1-415, 1-416, 1-417, 1-418, 1-419, 1-420, 1-421, 1-422, 1-423, 1-424, 1-425, 1-426, 1-427, 1-428, 1-429, 1-430, 1-431, 1-432, 1-433, 1-434, 1-435, 1-436, 1-437, 1-438, 1-439, 1-440, 1-441, 1-442, 1-443, 1-444, 1-445, 1-446, 1-447, 1-448, 1-449, 1-450, 1-451, 1-452, 1-453, 1-454, 1-455, 1-456, 1-457, 1-458, 1-459, 1-460, 1-461, 1-462, 1-463, 1-464, 1-465, 1-466, 1-467, 1-468, 1-469, 1-470, 1-471, 1-472, 1-473, 1-474, 1-475, 1-476, 1-477, 1-478, 1-479, 1-480, 1-481, 1-482, 1-483, 1-484, 1-485, 1-486, 1-487, 1-488, 1-489, 1-490, 1-491, 1-492, 1-493, 1-494, 1-495, 1-496, 1-497, 1-498, 1-499, 1-500, 1-501, 1-502, 1-503, 1-504, 1-505, 1-506, 1-507, 1-508, 1-509, 1-510, 1-511, 1-512, 1-513, 1-514, 1-515, 1-516, 1-517, 1-518, 1-519, 1-520, 1-521, 1-522, 1-523, 1-524, 1-525, 1-526, 1-527, 1-528, 1-529, 1-530, 1-531, 1-532, 1-533, 1-534, 1-535, 1-536, 1-537, 1-538, 1-539, 1-540, 1-541, 1-542, 1-543, 1-544, 1-545, 1-546, 1-547, 1-548, 1-549, 1-550, 1-551, 1-552, 1-553, 1-554, 1-555, 1-556, 1-557, 1-558, 1-559, 1-560, 1-561, 1-562, 1-563, 1-564, 1-565, 1-566, 1-567, 1-568, 1-569, 1-570, 1-571, 1-572, 1-573, 1-574, 1-575, 1-57

אם הגזירה היא חזקה, אין שהספר האחרון (החיצוני) הוא גזירי,
ונראה ש' ו' מ' - לפי "גזר" אל high למסל שלפניו.
(* גזרין שיהי ס מספר להחליף קמח אחר ולפחות (היה הגזירה גזולה או גזירה קמח)

```
int left = 0, right = a.length - 1;
while (left < right) {
    if (a[left] + a[right] == num) return true;
    else if (a[left] + a[right] > num) right--;
    else left++;
}
return false;
```


9.7 סיבוכיות מקום

מה?

סיבוכיות מקום היא סדר הגודל של מספר תאי הזיכרון שמתצטבר בזמן ריצתו של קוד.
 כמות זיכרון: $O(1)$
 קוד מוגבל רק שני גאונים - חזיון וזמן סיבוכיות המקום היא קבועה - כמות זיכרון $O(1)$.
 int i, n;
 for (i=0; i<n, i++)
 System.out.println(i);

דוגמאות

דוגמא 2 - $O(n)$: סיבוכיות הזיכרון של קוד
 int n;
 int[] a = new int[n];
 הקוד הזה הוא סיבוכי זיכרון $O(n)$ כי נוצר מספר קבוע של תאים.

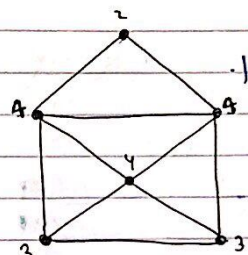
דוגמא 3 - $O(1)$:
 public static void printArray(int[] a) {
 for (int i=0; i<a.length; i++)
 System.out.println(a[i]);
 }
 אנו מרחיבים מערך "זיכרון" וזמן המקום הוא $O(1)$ כי יש לנו קבוע של תאים.

קבוצת שיטות מקבול - לשיטה, הסיבוכיות היא "קבועה" - $O(1)$.
 ההגיון הזה נובע מהגיון שבו קבוצת הסיבוכיות הזמן של שיטות קבועה היא $O(\log_2 n)$ -
 שם לא התחשבו קבוצת שבו לזמן להכניס את המספרים לזמן ההגיון קבוצת הסיבוכיות קבועה
 כמו פרימטר, וזמן לא זמן שלב לזמן אחר - כי לא קרה מחזוריות, אלא
 אולי שיטה קבועה של המעגל נכנסת - ולכן זה לא מעניין. נגזרים רק אלה שהיה קבוע
 הסיבוכיות הזמן, ולעשה כי זה סוגו נחשב סיבוכיות זמן.

סיכום

- קבוצת סיבוכיות מקום נקבעת רק לזמן הזיכרון הנדרש שהאלגוריתם מוציא אחר.
- ומשמש קבוצת קבוצת הזיכרון.
- הסיבוכיות הזמן נקבעת על ידי מספר הזיכרון של סיבוכיות הזמן.
- מספר סיבוכיות הזמן חשבו ממוצע קבוצת סיבוכיות של אלגוריתם.

אווילר - המילטון



מסלול שחוזר בכל הצמתים נקרא מסלול המילטון.
 מסלול שחוזר בכל הקצוות נקרא מסלול אוילר.
 במידה ובסוף המסלול נחזור לצומת בה התחלנו
 נקרא למסלול מועדף המילטון או מועדף אוילר.

אווילר

משורת קולמוס אחרת - קבוצת הווי קבוצת מסלול אוילר.
 דרגה של צומת - מספר הקצוות היוצאות מאותו צומת.
 משפט אוילר

- קבוצת הווי קבוצת מסלול אוילר.
- אם הדרגה של כל צומת קבוצת הווי קבוצת מסלול אוילר.
- כלומר: הווי "מלא" אוילר.
- אם יש שני צומת קבוצת הווי קבוצת מסלול אוילר, אז יש מסלול אוילר.
- אילו כל שני הצומת הם מדרגה זוגית - אז יש מסלול אוילר.
- קבוצת מסלול אוילר - אלא מדרגות קבוצת מסלול אוילר.
- ומסלול קבוצת הווי קבוצת מסלול אוילר.
- כמות מדרגות "מלא" אוילר.