שאלה 1:

הרצנו את הקוד בפיתון ורשמתי את הקוד והתוצאות.

https://github.com/1shaked/year_3/blob/main/algo2/HW1/quicksort.py

--------A--------

A: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

left: []

middle: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

right: []

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

--------B--------

B: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

left: [0, 1, 2, 3, 4]

middle: [5]

right: [6, 7, 8, 9]

left: [0, 1]

middle: [2]

right: [3, 4]

left: [0]

middle: [1]

right: []

left: [3]

middle: [4]

right: []

left: [6, 7]

middle: [8]

right: [9]

left: [6]

middle: [7]

right: []

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

--------C--------

C: [1, 0, 2, 1, 0, 2, 1, 0, 2, 1]

left: [1, 0, 1, 0, 1, 0, 1]

middle: [2, 2, 2]

right: []

left: []

middle: [0, 0, 0]

right: [1, 1, 1, 1]

left: []

middle: [1, 1, 1, 1]

right: []

[0, 0, 0, 1, 1, 1, 1, 2, 2, 2]

--------D--------

D: [5, 6, 3, 8, 1, 10, -1, 12, -3, 14]

left: [5, 6, 3, 8, 1, -1, -3]

middle: [10]

right: [12, 14]

left: [5, 6, 3, 1, -1, -3]

middle: [8]

right: []

left: [-1, -3]

middle: [1]

right: [5, 6, 3]

left: []

middle: [-3]

right: [-1]

left: [5, 3]

middle: [6]

right: []

left: []

middle: [3]

right: [5]

left: [12]

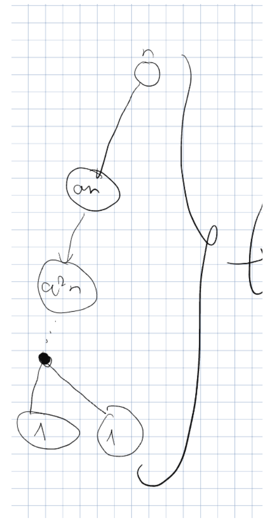
middle: [14]

right: []

[-3, -1, 1, 3, 5, 6, 8, 10, 12, 14]

שאלה 2:

אוקיי אז נוכיח, שאם אנחנו מניחים אורך t, ותיארנו באוויר את עץ הריקורסיה אזי מתקיים



$$a^t \cdot n = 1$$

$$\log_2 \textit{ on both sides} : \log_2(a^t \cdot n) = 0$$

$$\log a^t + \log n \ = 0$$

$$t \cdot \log a \ + \log n = 0$$

$$t = \frac{-\log n}{\log a}$$

עכשיו יש עלינו להבין למה זה גם האורך המינימלי,

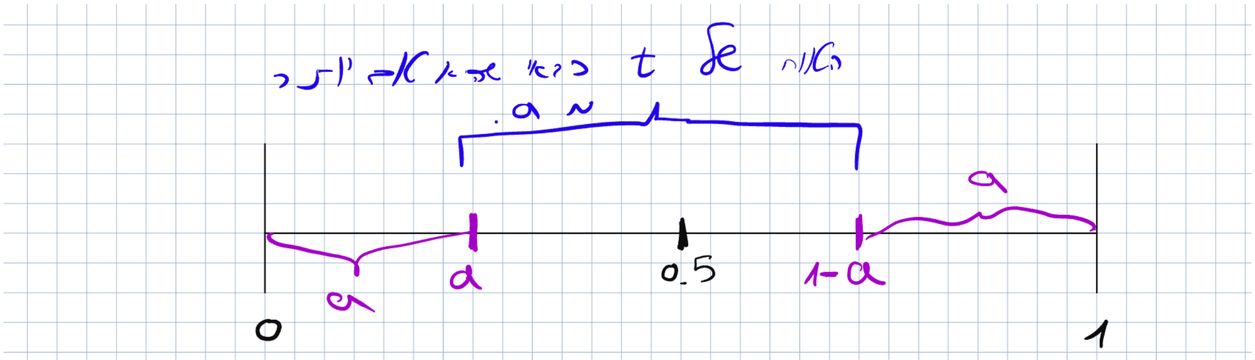
בעצם אנחנו בוחרים מספר שהוא קטן או שווה לחצי, שזה אומר שאנחנו מקטינים את הגודל של הרקורסיה. כמה שנקטין את המערך בכל שלב של הרקורסיה נצטרך פחות שלבים על מנת להגיע לאיבר אחד (שזה סוף הרקורסיה).

עכשיו אותו דבר נכון רק בצורה הפוכה לצד השני של הרקורסיה, שזה בעצם ככל שהצד השני של הרקורסיה גדול יותר ידרשו יותר שלבים על מנת להגיע לאיבר יחיד שהוא סוף הרקורסיה.

סעיף ב:

על מנת שהחלוקה תהיה פחות מאוזנת צריך שהממוצע של randomaztion partion ייתן בממוצע t כך ש t<a או t>1-a

מה הסיכוי שבעצם החלוקה תהיה כזאת



אבל השטח הינו 1-2a שזה בעצם הסיכוי להיות עם פרופורציה גדולה יותר.

מכיוון שהסיכוי הוא אחיד להיות בכל אחד מהסתברויות.

שאלה 3

בהינתן שנתון לנו k ו המערך באורך n.

מה שנצטרך לעשות זה למצוא את החלקים שעלינו לחלק את המערך אליו על מנת שנדע את ה k סטטיסטי שלו.

נבין איך לעשות את זה .

ראשית ניקח את אורך המערך n ונחלק אותו ב k (מניחים שהוא מתחלק שלם או קרוב לקח)

ניקח את הססטי האמצעי באותו מערך שהוא

$$a = \frac{k-1}{2} \cdot \frac{n}{k}$$

עכשיו ננסה למצוא את אותו סטטיסטי על a,

הוכחנו בכיתה שעל מנת למצוא סטטיסטי u צריך O(n)

עכשיו יהיה לנו מערך שיש לו את הססטיסטי ה a ממוין, שזה אומר שהוא נמצא במקום הנכון ואותו מצאנו, (כמובן נשמור את התוצאה)

שזה אומר המערך מחולק ל

$$\frac{a}{k} \cdot n \, , \left(1 - \frac{a}{k}\right) \cdot n$$

עכשיו נפעיל שוב את אותה הפונקציה שתיארנו פה רק שהפעם המערכים שנפעיל עליהם את הפונקציה יהיו בגדלים

$$a \, , n - a$$

והסטטיסטים שהם יתבקשו למצוא יהיו הרשימה של הסטטיסטים הקטנים מ a,

$$\frac{k-1}{2}$$

ואת הרשימה של הסטטיסטים הגדולים מ a,

$$\frac{k-1-1}{2}$$

הורדנו אקסטרה אחד מכיוון שאת אותו שסטטיסטי מצאנו בריצה זו שהוא האמצעי.

עכשיו ניקרא ברקורסיה לשני הצדדים של המערך, עם אותם דרישות.

זמן הריצה למציאה כל סטטיסטי יורד בערך בממוצע בחצי בכל ריצה.

״ אז מציאת הסטטיסטי הראשון לקח לנו O(n) זמן ריצה, אבל מציאת ברקורסיה את כל צד יקח לנו

$$O(a) + O(n-a) = O(n)$$

בממוצע מכיוון שחלקנו את המערך.

מוצאת כפול סטטיסטים ב O(n) זמן וככה ממשיך לאורך log(k) פעמים. מכיוון שכל פעם אנחנו מכפילים את כמות הסטטיסטים שמצאנו ב O(n) של זמן. ״

ולכן סה״כ זמן הריצה הינו

$$O(n \cdot \log(k))$$