ברוכים הבאים לתרגול 12 ☺

שחר אנגל

shaharbel0@gmail.com

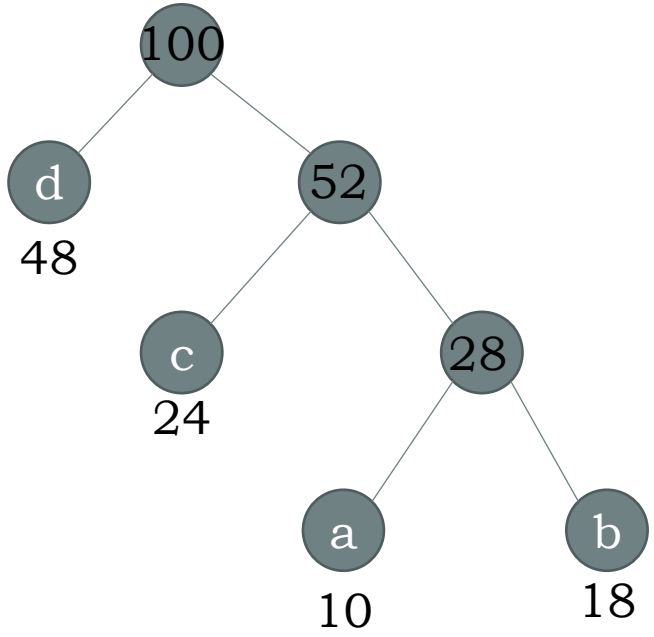
תרגול- ימי שני 14-16 וימי חמישי 13-15

נושא התרגול

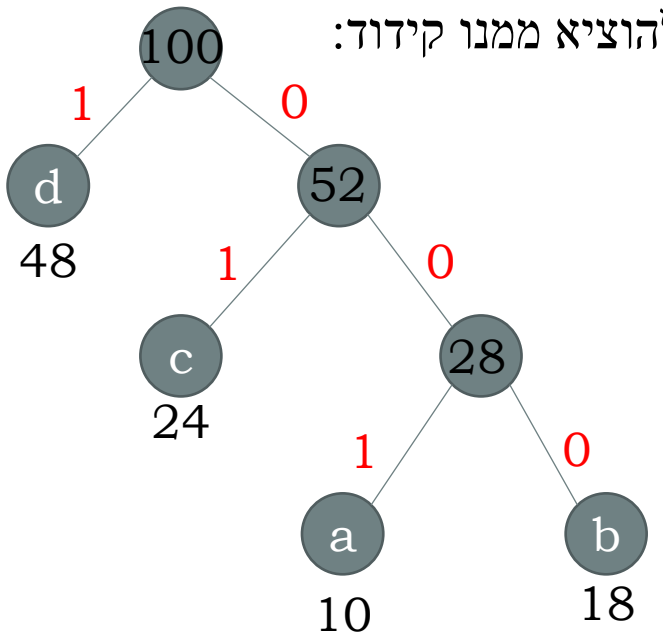
- קידוד הופמן

- **קידוד הופמן**
- מטרה: לקודד טקסט בשימוש במינימום ביטים ולהצליח לפענח אותו בחזרה בצורה מדויקת

- <u>לדוגמא</u>: a=10%, b=18%, c=24%, d=48%
- נרצה שככל שהתו מופיע יותר פעמים הוא יהיה קרוב לשורש העץ, וככל שהוא מופיע פחות הוא יהיה קרוב לעלים.
- בוא נבנה את העץ:



- **קידוד הופמן**
- קיבלנו עץ שניתן להוציא ממנו קידוד:



a = 001
b = 000
c = 01
d = 1

- כעת ניתן לחשב כמה זיכרון זה יתפוס:
- 3*10 + 3*18 + 2*24 + 1*48 = 180
- מה אומר ה-180 הזה?
  - אם יש לנו מחרוזת באורך x שאלו האחוזים של הקידוד אז כל תו יתפוס 1.8 עבור כל תו בממוצע

- **קידוד הופמן**
- איך ניתן לממש את העץ?
- בכמה דרכים:
- **<u>דרך א'- סטטית:</u>**
- ברגע שיש 4 תווים אנו יודעים מה יהיה גודל העץ. איך?
  - n-1 קודקודים פנימיים + n קודקודים של התווים = 2n-1
  - כך ניתן פשוט ליצור מטריצה בגודל הרלוונטי

- נחזור לדוגמא מקודם: a=10%, b=18%, c=24%, d=48%

- כדי לא לקבל קידודים שונים, נגדיר שהבן עם הערך הקטן יותר יהיה הבן השמאלי.
- כמובן שלא נשכח לסמן בכל סוף איטרציה במי השתמשנו כבר

				אבא	בן ימני 0	בן שמאלי 1
1	a	10				
2	b	18				
3	c	24				
4	d	48				
5						
6						
7						

- **קידוד הופמן**
- למה לא כל המטריצה מלאה?
  - נכון מאוד, לכל העלים אין בנים, ולשורש אין אבא
- האם המטריצה היא עץ?
  - כן, במקום לייצר עם Node וקשרים אחרים, פישטנו את זה. כל אחד יודע מי האבא שלו, מי בן שמאלי וכו'
- מה הסיבוכיות?
  - כמה איטרציות היו?
    - n-1
  - מה עושים בכל איטרציה?
    - מחפשים 2 תווים מינימלים- n+logn-2
    - מקצים להם אבא- o(1)
    - מציבים לאבא את הבנים- o(1)
    - מציבים לבנים את האבא- o(1)
    - סוכמים אחוזים לאבא- o(1)
    - מסמנים בנים כ'משומשים'- o(1)
  - לכן סה"כ נקבל- o(n^2)
- איך ניתן לשחזר את הקידוד?
  - נשאל כל תו מי אבא שלו, נשאל את האבא האם התו בן ימני או שמאלי, ונמשיך כך עד שנגיע לשורש.

			אבא	בן ימני 0	בן שמאלי 1
1	a	10	5		
2	b	18	5		
3	c	24	6		
4	d	48	7		
5		28	6	2	1
6		52	7	5	3
7		100		6	4

- **קידוד הופמן**
- <u>פסאודו קוד של **דרך ב'**:</u>
- לפני שנתחיל, מה הפעולה שחוזרת על עצמה הכי הרבה פעמים?
  - חיפוש מינימום
- אם נחפש עכשיו כל כך הרבה פעמים זה יעלה לנו ביוקר, אז במה נוכל להעזר?
  - ערמת מינימום- הממומשת ע"י תור עדיפויות

- <u>Huffman(C):</u>
- o(1) n = |C|
- o(n) Q = C
- for i=1 to n-1:
  - o(1) z = allocate Node()
  - o(logn) x = z.left = Extract_Min(Q)
  - o(logn) y = z.right = Extract_Min(Q)
  - o(1) f(z) = f(x) + f(y)
  - o(logn) insert(Q,z)
- return Extract_Min(Q)

סיבוכיות?
o(nlogn)
הצלחנו להוריד את הסיבוכיות כאשר שינינו את מבנה הנתונים

- **קידוד הופמן**
- **<u>דרך ג':</u>**
- נניח ונותנים לנו את התווים ממויינים. נוכל לשפר עוד את הסיבוכיות אם נשתמש במיון וכך השליפה מהתור תהיה ב-o(1):
- נשתמש ב-2 תורים כך:

- כמה איטרציות עשינו?
  - n-1
- בכל איטרציה איזה פעולות עשינו?
  - השוואה, שליפה והכנסה לתור
- זאת אומרת שהסיבוכיות בסה"כ היא o(n)

אז מה צריך לתכנת?

- כל מה שדיברנו עליו היום ☺
- קידוד הופמן
  1. בעזרת מטריצה
  2. בעזרת ערימת מינימום
  3. בעזרת 2 תורים (במידה וזה ממויין)
- עבור כל אחת מהדרכים צריך להחזיר את הקידוד

בהצלחה ☺