

מבני נתונים – מבני נתונים בסיסיים דף תרגילים

שאלה 1

כתבו אלגוריתם המקבל מיקום של פרש על לוח שחמט שגודלו $n \times n$. ומחזיר בכל תא במערך את אורך המסלול הקצר ביותר של הפרש אליו.

שאלה 2

רשימה מקושרת יכולה להסתיים במצביע null (ואז נקרא לה רשימה קווית) או במצביע לאיבר קודם ברשימה (ואז נקרא לה שבלול).
כתבו אלגוריתם המקבל מצביע לרשימה מקושרת (המכילה לפחות איבר אחד) ומחזיר true אם ורק אם הרשימה היא שבלול. זמן הריצה הוא לינארי באורך הרשימה.

שאלה 3

תארו כיצד לממש "מחסנית מינימום" – מחסנית שחוץ מפעולות POP ו-PUSH, ממומשת עליה גם פעולת MIN המחזירה את הערך המינימלי במחסנית (רק מחזירה ולא מוחקת).
זמני הריצה של שלוש הפעולות יהיו קבועים.

שאלה 4

כתבו אלגוריתם המקבל תור והופך אותו. כלומר: האיבר הראשון יהפוך לאחרון, האיבר האחרון יהפוך לראשון וכו'.
מספר האיברים בתור, $n > 0$, ידוע.
האלגוריתם ישתמש ב- $O(1)$ זיכרון נוסף, וזמן הריצה שלו יהיה $O(n^2)$.

שאלה 5

נתונים שני מספרים עשרוניים גדולים מאוד x, y .
כל אחד מהמספרים נמצא במחסנית, כך שהספרה המשמעותית ביותר נמצאת בתחתית המחסנית.
כתבו אלגוריתם המקבל שתי מחסניות $S1, S2$ שבהן מאוחסנים המספרים x ו- y ומחזיר מחסנית $S3$ שבה נמצא הסכום $x + y$, (גם ב- $S3$ הספרה המשמעותית ביותר תהיה בתחתית המחסנית)

מותר להשתמש אך ורק בפעולות הבסיסיות המוגדרות על מחסנית.
בתום ריצת האלגוריתם $S1, S2$ תהיינה ריקות.
עליכם לכתוב את האלגוריתם בפסידוקוד.

מבני נתונים – מבני נתונים בסיסיים דף תרגילים - המשך

שאלה 6

כתבו אלגוריתם המקבל מצביע לעץ בינרי ובודק האם העץ הוא עץ בינרי שלם. כלומר האם כל הרמות של העץ מלאות.
זמן הריצה הנדרש של האלגוריתם $O(n)$.

שאלה 7

כתבו אלגוריתם רקורסיבי המקבל מצביע לעץ בינרי בעל n צמתים, ובודק האם העץ מקיים את תכונת הערמה: הערך בכל צמת לא קטן מהערכים בבניו.
סיבוכיות זמן הריצה הנדרשת היא $O(n)$.

שאלה 8

כתבו אלגוריתם רקורסיבי המקבל מצביע לעץ בינרי ומספר שלם חיובי k . האלגוריתם ידפיס את כל הצמתים ברמה k . הניחו שרמת השורש 0, רמת בניו 1 וכו'.

שאלה 9

הסבירו כיצד להשתמש באלגוריתם משאלה 8 כדי לממש BFS על עץ נתון. מה זמן הריצה של האלגוריתם במקרה הגרוע ביותר ובמקרה הטוב ביותר? מספר הצמתים בעץ הוא n .

מבני נתונים – מבני נתונים בסיסיים פתרונות דף התרגילים - למרצה

מבני נתונים בסיסיים – פתרון שאלה 1

האלגוריתם המצורף מקבל מיקום של פרש $(r1, c1)$ על לוח שח בגודל A (שגודלו A הוא מייצג לוח שח) את מספר הצעדים שעל פרש לעשות כדי להגיע מהמיקום $(r1, c1)$ לכל משבצת בלוח שהמערך מייצג. אם לא ניתן להגיע למשבצת מסוימת, הערך בתא המתאים במערך יהיה -1.

האלגוריתם מבצע BFS בעזרת תור.

ראשונה נכנסת לתור המשבצת בה ממוקם הפרש. התא ב- A בו נמצא הפרש מסומן ב-0 (כי צריך 0 מהלכים כדי להגיע אליו).

אחר-כך, כל עוד התור לא ריק, האלגוריתם מוציא מהתור קואורדינטות של משבצת ומכניס לתור את הקואורדינטות של כל המשבצות שניתן להגיע ממנה אליהן במהלך פרש יחיד.

השורות האדומות באלגוריתם מאתחלות מערכים שמחזיקים את המיקומים היחסיים של שמונה המשבצות אליהן ניתן להגיע במהלך יחיד של פרש.

השורות הכחולות מאתחלות את המערך A .

השורות בוורוד מטפלות במשבצת בה ממוקם הפרש- מכניסות אותה למערך, ושמות בתא המתאים ב- A את הערך 0.

השורות בירוק עוברות על כל שמונה המשבצות שניתן להגיע אליהן, במהלך פרש יחיד, מן המשבצת הנוכחית. לכל אחת מהן האלגוריתם בודק האם היא בתוך הלוח והאם הוא עדיין לא ביקר בה (הערך בתא המתאים ב- A עדיין -1), ואם הכל בסדר הוא מסמן את התא המתאים במערך A ומכניס אותה לתור.

Knight(r1,c1,A,n)

dr=[2,1,-1,-2,-2,-1,1,2]
dc=[1,2,2,1,-1,-2,-2,-1]

for r←0 to n-1 do
 for c←0 to n-1 do
 A[r,c]←-1

enqueue(q,[r1,c1])
A[r1,c1]←0

while not empty(q) do
 [r,c]←dequeue(q)
 for i←0 to 7 do
 if 0≤r+dr[i]<n and 0≤c+dc[i]<n
 and A[r+dr[i],c+dc[i]]=-1
 then A[r+dr[i],c+dc[i]] ← A[r,c]+1
 enqueue(q,[r+dr[i],c+dc[i]])

מבני נתונים בסיסיים – פתרון שאלה 2

האלגוריתם ירוץ על הרשימה בעזרת שני מצביעים. מצביע אחד איטי שיזוז איבר אחד בכל איטרציה, ומצביע שני מהיר שיזוז שני איברים בכל איטרציה. אם המצביע המהיר מגיע ל-null הרשימה קוויית. אם המצביע המהיר נפגש עם המצביע האיטי הרשימה שבלול.

```
snail (p)  
slow ← p  
fast ← next[p]  
while fast ≠ slow and fast ≠ null do  
    slow ← next[slow]  
    fast ← next[fast]  
    if fast ≠ null  
        then fast ← next[fast]  
return fast = slow
```

מבני נתונים בסיסיים – פתרון שאלה 3

מבנה הנתונים יורכב משתי מחסניות. שתי המחסניות יכילו אותו מספר איברים. הראשונה תכיל את אברי המחסנית מינימום. השנייה תאחסן בראש המחסנית את האיבר המינימלי. תחילה נגדיר פעולת עזר המחזירה את האיבר שבראש המחסנית - top

top(s) $x \leftarrow \text{pop}(s)$

push(s, x)

return x

push(s, x)

push(s[1], x)

if empty(s[2]) or $x < \text{top}(s[2])$

then push(s[2], x)

else push(s[2], top(s[2]))

pop(s)

pop(s[2])

return pop(s[1])

min(s)

return top(s[2])

מבני נתונים בסיסיים – פתרון שאלה 4

```
enqueue(q, dequeue(q))  
for i ← 1 to n-2 do  
    x ← dequeue(q)  
    for j ← 1 to n-1-i do  
        enqueue(q, dequeue(q))  
    enqueue(q, x)  
    for j ← 1 to i do  
        enqueue(q, dequeue(q))
```

מבני נתונים בסיסיים – פתרון שאלה 5

```
c ← 0
while (not empty(s1) or not empty(s2)) do
    if empty(s1) // הספרה מהמחסנית הראשונה
    then x ← 0
    else x ← pop(s1)
    if empty(s2) // הספרה מהמחסנית השנייה
    then y ← 0
    else y ← pop(s2)
    push(s3, (x+y+c) mod 10) // חיבור
    c ← (x+y+c) / 10 // חישוב הנשא
if c > 0 then push(s3, c) // הנשא האחרון
while not empty(s3) do // שלוש הלולאות הופכות את הסדר
    push(s2, pop(s3))
while not empty(s2) do
    push(s1, pop(s2))
while not empty(s1) do
    push(s3, pop(s1))
```


מבני נתונים בסיסיים – פתרון שאלה 6

האלגוריתם מחזיר שני ערכים: האם העץ שלם? גובה העץ.
עץ הוא שלם אם ורק אם שני התת-עצים שלו הם שלמים ובאותו גובה.

```
isComplete(t)  
if t=nil  
then return [true,-1]  
[ansl,hl]←isComplete(left[t])  
[ansr,hr]←isComplete(right[t])  
ans← ansl and ansr and hl=hr  
h←MAX(hl,hr)+1  
return [ans,h]
```

מבני נתונים בסיסיים – פתרון שאלה 7

```
check(t)
if t=nil
then return true
if parent(t)≠nil and key(t)>key(parent(t))
then return false
return check(left(t)) and check(right(t))
```

מבני נתונים בסיסיים – פתרון שאלה 8

```
printLevel(t,k)
if t!=nil
then if k=0
    then print(key(t))
    else printLevel(left(t), k-1)
        printLevel(right(t), k-1)
```

מבני נתונים בסיסיים – פתרון שאלה 9

נמצא את מספר הרמות של העץ ולכל רמה נקרא לאלגוריתם מהשאלה הקודמת.
במקרה הגרוע זמן הריצה הוא ריבועי: $O(n^2)$. במקרה הטוב ביותר זמן הריצה לינארי $O(n)$.