



תור Queue דף עבודה מס' 2

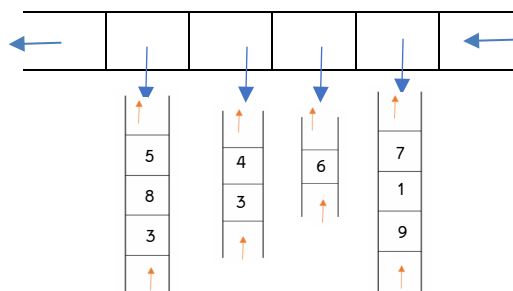
עבודה 1

- א. כתבו פעולה בשם `SumQueue / sumQueue` שתקבל כפרמטר תור ותחזיר את הסכום של האיבר שנמצא בראש התור והאיבר שנמצא הסוף התור. אם יש בתור רק איבר אחד, הרי שהוא האיבר שבראש התור וגם האיבר שבסופו, ולכן הוא יוחזר כשהוא מוכפל ב-2. אם התור ריק יוחזר 0. בסיום הפעולה התור יישאר ללא שינוי.
- ב. נתון תור `qQue` שכל איבר בו הוא תור של מספרים שלמים (ראה איור), ונתונה הפעולה הרקורסיבית `sod`. עקבו בעזרת טבלת מעקב אחר זימון הפעולה `sod(qQue, 4) / Sod(qQue, 4)` ורשמו מהו הפלט שמוחזר. על טבלת המעקב להכיל עמודה לתור `qQue` ועמודה למה שמחזירה הפעולה `sumQueue` בנוסף לעמודות הרגילות. יש להציג את התורים בטבלת המעקב, כפי שמחזירה הפעולה `toString` למבנה הנתונים: [תור-1, תור-2, ..., תור-n]

| Java | C# |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| טענת כניסה: הפעולה מקבלת תור של תורים ומספר טענת יציאה: _____ | |
| <pre> public static int sod (Queue<Queue<Integer>> qQue , int n) { if (n == 0) return 0; Queue<Integer> qu1 = qQue.remove(); int y = sumQueue(qu1); qQue.insert (qu1); return y + sod (qQue, n-1); } </pre> | <pre> public static int Sod (Queue<Queue<int>> qQue , int n) { if (n == 0) return 0; Queue<int> qu1 = qQue.Remove(); int y = SumQueue(qu1); qQue.Insert (qu1); return y + Sod (qQue, n-1); } </pre> |

qQue : [[5,8,3], [4,3],[6],[7,1,9]]

תור של תורים



שאלה 2

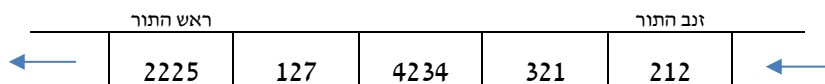
לפניכם 2 פעולות חיצוניות:

טענת כניסה: הפעולה מקבלת תור לא ריק, המכיל מספרים שלמים.

טענת יציאה: הפעולה מחזירה ...

| Java | C# |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> public static int sod1(Queue<Integer> q) { int i = q.remove(); int result = i; if (!q.isEmpty()) { int j = sod1(q); if (result > j) result = j; } q.insert(i); return result; } public static int sod2(int i) { if (i == 0) return 0; int a = i % 10; int b = sod2(i / 10); if (a > b) return a; return b; } </pre> | <pre> public static int Sod1(Queue<int> q) { int i = q.Remove(); int result = i; if (!q.IsEmpty()) { int j = Sod1(q); if (result > j) result = j; } q.Insert(i); return result; } public static int Sod2(int i) { if (i == 0) return 0; int a = i % 10; int b = Sod2(i / 10); if (a > b) return a; return b; } </pre> |

נתון התור myQueue מטיפוס Queue<int> / Queue<Integer>



- מה יחזירה הזימון Sod1(myQueue)? רשמו את המעקב. רשמו את התור המתקבל בתום המעקב. ציינו את ראש התור ואת סוף התור.
- מה מבצעת הפעולה Sod1(queue) בעבור תור queue לא ריק מטיפוס Queue<int>/Queue<Integer>?
- מה יחזיר הזימון Sod2(17852)? רשמו את המעקב.
- מה מבצעת הפעולה Sod2(k) בעבור k גדול מ-0 מטיפוס שלם?
- מה מבצעת הפעולה Sod2(Sod1(queue)) בעבור תור queue לא ריק מטיפוס Queue<int> / Queue<Integer> המכיל רק מספרים שלמים וגדולים מ-0?

עבודה 3

תור "שווה סכומים", הוא תור המכיל מספר אי זוגי של איברים. כך שכל שני איברים קיצוניים (ראשון+אחרון, שני + לפני אחרון וכו'). שווים בסכומם לאיבר האמצעי בתור.

לדוגמא – התור הבא הוא תור "שווה סכומים".

| | | | | | | | | | | | | |
|---|----|---|----|----|---|----|----|----|----|----|---|---|
| ← | 18 | 3 | 15 | 13 | 4 | 25 | 21 | 12 | 10 | 22 | 7 | ← |
|---|----|---|----|----|---|----|----|----|----|----|---|---|

האיבר האמצעי בתור הוא 25. וכל זוג איברים קיצוניים שווה ל – 25.

$$\begin{aligned} 25 &= 18+7 \\ 25 &= 3+22 \\ 25 &= 15+10 \\ 25 &= 13+12 \\ 25 &= 4+21 \end{aligned}$$

כתבו פעולה בשם EqualSums/equalSums המקבלת תור ומחזירה אמת אם התור הוא "שווה סכומים" ושקר אחרת.

עבודה 4

א. תור לא מסודר הוא תור של מספרים שלמים אשר חוזרים על עצמם באופן אקראי.

לדוגמא התור שלפניכם הוא תור לא מסודר.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ← | 2 | 1 | 2 | 4 | 7 | 4 | 4 | 7 | 9 | 2 | 7 | 9 | 1 | 5 | 4 | ← |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

"תור מסודר" הינו תור המקיים את התנאים הבאים:

- ✓ התור אינו ריק.
- ✓ ערכי התור המכילים מספרים שלמים המקובצים ברצף.
- ✓ אין שני רצפים של אותו מספר.

תור מסודר:

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ← | 2 | 2 | 2 | 1 | 1 | 4 | 4 | 4 | 4 | 7 | 7 | 7 | 9 | 9 | 5 | ← |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

א. כתבו פעולה בשם CountX / countX המקבלת מספר שלם ותור של מספרים שלמים. הפעולה תחזיר את מספר מופעיו של המספר בתור. בסיום יישאר התור ללא מופעיו של מספר זה.

- ב. כתבו פעולה המקבלת תור לא מסודר של מספרים שלמים. על הפעולה לבנות ולהחזיר תור מסודר באופן הבא:
- לכל מספר המופיע בתור יש למצוא את כל המספרים הזוהים לו, ולהכניס אותם בצורה רציפה ומסודרת לתור החדש.
- על הפעולה להחזיר את התור המסודר.
- על הפעולה להיעזר בפעולה שכתבת בסעיף א'.
- מה יש לשנות בקוד אם נדרוש שבסיום יישאר התור ללא שינוי?

ג. מהי יעילות הפעולה שכתבתם? נמקו

עבודה מחכימה