

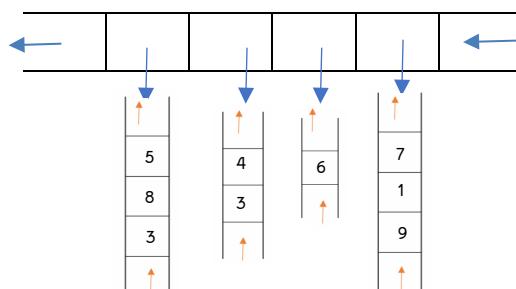


תור Queue דף עבודה מס' 2

1. מילוי Queue

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

| Java | C# |
|--|--|
| // // טענת כניסה: הפעולה מקבלת תור של תורים ומספר טענת יציאה: | 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); } 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); } |
| qQue : [[5,8,3], [4,3],[6],[7,1,9]] | תור של תורים |



2 סעיפים

לפניכם 2 פעולות חיצוניתות :

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; }</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; }</pre> |
| <pre>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 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 Queue<int> / Queue<Integer> myQueue

| ראש התור | 2225 | 127 | 4234 | 321 | 212 | בסוף התור |
|----------|------|-----|------|-----|-----|-----------|
| ← | | | | | ← | |

א. מה יחזיר היזיון Sod1(myQueue) ? רשמו את התור המתקבל בתום המעקב. ציינו את ראש התור ואת סוף התור.

ב. מה מבצעת הפעולה Sod1(queue) בעברית תור queue לא ריק מטיפוס ?

ג. מה יחזיר היזיון Sod2(17852) ? רשמו את המעקב.

ד. מה מבצעת הפעולה Sod2(k) בעברית k גדול מ – 0 מטיפוס שלם ?

ה. מה מבצעת הפעולה Sod2(Sod1(queue)) בעברית תור queue לא ריק מטיפוס / Queue<Integer> המכיל רק מספרים שלמים וגדולים מ – 0 ?

3 מילים

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

לדוגמא – התור הבא הוא תור "שווה סכומים".

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

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

$$\begin{aligned} \text{(האייר הראשון והאייר האחרון).} & 25 = 18+7 \\ \text{(האייר השני והאייר שלפניו אחרון).} & 25 = 3+22 \\ & 25 = 15+10 \\ & 25 = 13+12 \\ & 25 = 4+21 \end{aligned}$$

כתבו פעולה בשם EqualSums/equalSums כתבו פעולה בשם EqualSums/equalSums וSKU אחרת.

4 מילים

א. תור לא מסודר הוא תור של מספרים שלמים אשר חוזרים על עצם 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 המכיל את התנאים הבאים:

- ✓ התור אינו ריק.
- ✓ ערכי התור המכילים מספרים שלמים המקבצים ברכף.
- ✓ 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 / count המקבלת מספר שלם ותור של מספרים שלמים. הפעולה תחזיר את מספר מופיעו של המספר בתור. בסיום יישאר התור ללא מופיעו 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 הזהים לו,
 ולהכניס אותם לצורה רציפה ומסודרת לתור החדש.

על הפעולה להחזיר את התור המסודר.

על הפעולה להיעזר ב פעולה שכותבת בסעיף א'.

מה יש לשנות בקוד אם נדרש שבסיום יישאר התור ללא שינוי?

ג. מהי יעילות הפעולה שכותבתם? נמקו

אנו, מתיים