Дизайн и Анализ на Алгоритми

Примерни решения на малко контролно № 2

50 т. **Задача 1.** За всеки числен масив $A[1, \ldots, n]$ казваме, че A е *битоничен*, ако

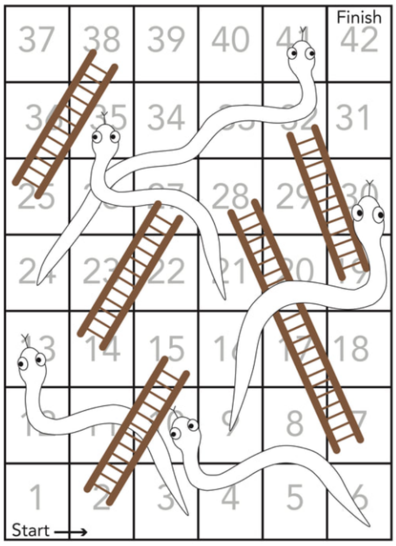
$$A[1] < A[2] < \ldots A[k-1] < A[k] > A[k+1] > \ldots > A[n-1] > A[n].$$

По подаден числен масив $A[1, \ldots, n]$ да се намери дължината на най-дългата **подредица**, която образува битоничен масив.
Забележка: Алгоритъмът **трябва** да е съставен по схемата **динамично програмиране** и да се напише **рекурсивната декомпозиция**.

20 т. *Бонус:* Модифицирайте алгоритъма, така че да върнете самата най-дълга подредица, а не само нейната дължина.

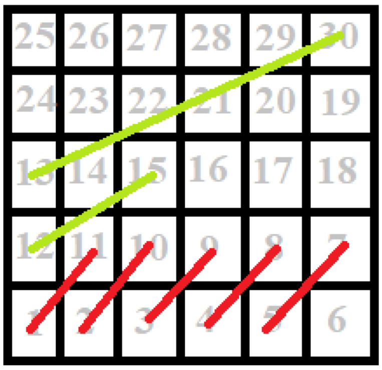
Решение: Понеже неравенствата в условието са строги, то може да разгледаме най-дългата подредица, чийто най-голям елемент е $A[i]$ за всяко i. Тя се образува като вземем най-дългата нарастваща подредица завършваща в елемента $A[i]$ + най-дългата намаляваща подредица започваща в елемента $A[i]$ и извадим 1 (понеже броим елемента $A[i]$ два пъти). На упражнение сме решавали задачата за най-дълга нарастваща подредица (Упр. 11, зад. 3). Модифицирането на кода за намирането на най-дълга намаляваща подредица е тривиално (знака от $<$ става на $>$). За да намерим отговора просто итерираме $\forall i \in \{1, \ldots, n\}$ и взимаме максимума от $(LIS[i] + LDS[i] - 1)$.
Ако искаме да модифицираме алгоритъма, така че да върнем самата подредица, която образува най-дълъг битоничен масив, то намираме за кой индекс $i \in \{1, \ldots, n\}$ е изпълнено $(LIS[i] + LDS[i] - 1)$ да е максимално. След което вървим от този индекс наляво в масива $LIS[1, \ldots, n]$, като търсим първият елемент, чиято стойност е $A[i] - 1$, след което първият елемент, чиято стойност е $A[i] - 2$, и така докато не стигнем до елемент, чиято стойност е 1 (понеже всеки елемент сам по себе си е нарастваща подредица). Аналогично правим и за масива $LDS[1, \ldots, n]$, само че вървим надясно.

50 т. **Задача 2.** Дадена е дъска на играта "Змии и стълби". Намерете минималния брой хвърляния на стандартно зарче, така че от началното поле да се достигне до финалното. Когато играчът попадне в поле, от което започва стълба, пионката му се мести в клетката, до която стълбата води. В случай че играчът попадне в поле, в което се намира главата на змия, пионката му се придвижва до полето, съдържащо опашката на змията.
Достатъчно е да опишете алгоритъма на български език, **ясно и недвусмислено** – не се изисква псевдокод. Дайте кратка аргументация за коректност и направете анализ на сложността по време и по памет. Можете наготово да използвате алгоритми изучавани на лекции.



Пример: За дъската на фигурата, играчът може да достигне от началното поле до крайното за 4 хода, например последователно хвърляйки $(6, 6, 6, 2)$.

Решение: Ще конструираме ориентиран нетегловен граф $G = (V, E)$ по следния начин: Множеството от върховете V на графа е клетките на подадената ни дъска. За всеки връх слагаме ребро (i, j), като $j \in \{i+1, \ldots, i+6\}$ и j е по-малко от стойността на най-голямата клетка от дъската. Особеността е, че на дадено ребро (i, j), ако на позиция j започва стълба, то не добавяме реброто (i, j), а ребро започващо от i до клетката в която води стълбата, започваща на поле j. Тоест ако на поле j има стълба, която води към клетка k, то не добавяме реброто (i, j), а добавяме единствено реброто (i, k). Аналогично ако срещнем главата на змията, то добавяме ребро от клетка i към полето на което се намира опашката на змията. Това е така, защото по условие ние не избираме дали ще се качим по стълбата или ще слезем по змията, а това е задължително! След като сме конструирали графа, използваме алгоритъма BFS, който е изучаван на лекции и намираме най-късият път от началното поле до финалното, което съответства на минималния брой хвърляне на зарчета. Задачата може и да се реши, като графът е тегловен и всяко ребро е с тегло 1, след което да се използва алгоритъма на $Dijkstra$, но това ще доведе до по-лоша асимптотична сложност. Груба грешка е при конструиране на графа да се слагат ребра (i, j) за стълба или змия, чиято тежест е 0. Това е невалидно понеже така си избираме дали да се качим на стълба, което не е по правилата на играта и може да доведе на намиране на решение, което не е валидно. На фигурата по-долу със зелено са илюстрирани стълбите, а с червено са илюстрирани змиите. Ако реброто $(12, 15)$ е с тежест 0, то няма да се качим по нея стълба, а ще се качим по стълбата $(13, 30)$, което е невалиден ход.



Фигура 1: Контрапример