

СОРТИРАНЕ И ТЪРСЕНЕ
КОНТРОЛНО № 2 ПО “ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ”
ЗА СПЕЦИАЛНОСТ “КОМПЮТЪРНИ НАУКИ”, 2. КУРС, 1. ПОТОК
(СУ, ФМИ, ЛЕТЕН СЕМЕСТЪР НА 2018 / 2019 УЧ. Г.)

Задача 1. За всяка от описаните ситуации преценете каква сортировка е най-подходяща, като посочите само един от предложените отговори.

- 1) Да се сортира двусвързан списък на място, тоест с количество допълнителна памет $\Theta(1)$.
а) Сортиране чрез броене. б) Сортиране чрез сливане. в) Пирамидално сортиране.
г) Метод на бройните системи. д) Бързо сортиране. е) Сортиране чрез пряк избор.
- 2) Да се сортира възможно най-бързо масив от бройките братя и сестри на няколко души.
а) Сортиране чрез броене. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Бързо сортиране. е) Метод на бройните системи.
- 3) Да се сортира устойчиво и на място масив от дължините на няколко отсечки.
а) Сортиране чрез сливане. б) Сортиране чрез броене. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Топологично сортиране. е) Метод на бройните системи.
- 4) Да се сортира устойчиво и възможно най-бързо масив от теглата на няколко предмета.
а) Сортиране чрез сливане. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Сортиране чрез пряк избор. е) Сортиране чрез броене.
- 5) Да се сортира възможно най-бързо масив от плътностите на различни вещества.
а) Сортиране чрез броене. б) Сортиране чрез вмъкване. в) Метод на бройните системи.
г) Метод на мехурчето. д) Бързо сортиране. е) Сортиране чрез пряк избор.

Задача 2. За всяка от описаните ситуации преценете кой вид търсене е верен и най-бърз. Посочете само един от предложените отговори.

- 1) Търсим медианата на несортиран масив за най-малко време при най-лоши входни данни.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.
- 2) В несортиран масив $A[1...n]$, $A[1] \neq A[n]$, търсим някой индекс k , за който $A[k] \neq A[k+1]$.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.
- 3) Търсим индекса на елемент на сортиран списък по дадена стойност на елемента.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.
- 4) Търсим индекса на най-малък елемент в несортиран масив.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.
- 5) Търсим елемент на несортиран масив, по-голям от 25% от елементите, за най-малко време, усреднено по всевъзможните входни данни.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.

РЕШЕНИЯ

Задача 1. Верните отговори са подчертани.

- 1) Да се сортира двусвързан списък на място, тоест с количество допълнителна памет $\Theta(1)$.
а) Сортиране чрез броене. б) Сортиране чрез сливане. в) Пирамидално сортиране.
г) Метод на бройните системи. д) Бързо сортиране. е) Сортиране чрез пряк избор.

Обосновка: Сортирането чрез сливане копира данните, тоест не сортира на място. Сортирането чрез броене и методът на бройните системи имат специфични изисквания. Бързото и пирамидалното сортиране сравняват далечни елементи, поради което работят само върху масиви. Сортирането чрез пряк избор обхожда данните последователно, затова работи и върху списъци. От предложените отговори само то изпълнява всички изисквания.

- 2) Да се сортира възможно най-бързо масив от бройките братя и сестри на няколко души.
а) Сортиране чрез броене. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Бързо сортиране. е) Метод на бройните системи.

Обосновка: Броят братя и сестри е малко цяло неотрицателно число, затова е приложимо сортирането чрез броене, което има линейно време. Методът на бройните системи също има линейно време, но прави няколко паса — по един за всяка цифра, — докато сортирането чрез броене обхожда масива само веднъж. Другите методи са по-бавни.

- 3) Да се сортира устойчиво и на място масив от дължините на няколко отсечки.
а) Сортиране чрез сливане. б) Сортиране чрез броене. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Топологично сортиране. е) Метод на бройните системи.

Обосновка: Дължините на отсечките са дробни числа, затова сортирането чрез броене и методът на бройните системи не са приложими. Топологичното сортиране е за графи, не за числови масиви. Сортирането чрез сливане копира масива (не сортира на място). Пирамидалното сортиране работи на място, но е неустойчиво. Само методът на мехурчето отговаря на двете изисквания — устойчивост и работа на място.

- 4) Да се сортира устойчиво и възможно най-бързо масив от теглата на няколко предмета.
а) Сортиране чрез сливане. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Сортиране чрез пряк избор. е) Сортиране чрез броене.

Обосновка: Сортирането чрез броене не работи върху дробни числа (тегла на предмети). От другите алгоритми най-бързи са пирамидалното сортиране и сортирането чрез сливане. Но пирамидалното сортиране не е устойчиво. Остава сортирането чрез сливане.

- 5) Да се сортира възможно най-бързо масив от плътностите на различни вещества.
а) Сортиране чрез броене. б) Сортиране чрез вмъкване. в) Метод на бройните системи.
г) Метод на мехурчето. д) Бързо сортиране. е) Сортиране чрез пряк избор.

Обосновка: Плътностите на веществата са дробни числа, затова сортирането чрез броене и методът на бройните системи не са приложими. От останалите четири алгоритъма най-малка времева сложност ($n \log n$) има бързото сортиране; другите са квадратични.

Задача 2. Верните отговори са подчертани.

- 1) Търсим медианата на несортиран масив за най-малко време при най-лоши входни данни.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.

Обосновка: Последователното и двоичното търсене не могат да решат тази задача, а бързото търсене и алгоритъмът PICK могат. Обаче при най-лоши входни данни бързото търсене е по-бавно от PICK. Затова PICK е най-подходящият алгоритъм тук. Няма нужда да го съчетаваме с двоично търсене: това се прави в друг тип задачи.

- 2) В несортиран масив $A[1...n]$, $A[1] \neq A[n]$, търсим някой индекс k , за който $A[k] \neq A[k+1]$.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.

Обосновка: Приложими са само последователното и двоичното търсене. Двоичното търсене е по-бързото от двете, затова то е верният отговор. Може да се приложи така. Проверяваме средния елемент: ако той се различава от първия елемент, продължаваме търсенето в лявата половина; ако не — в дясната. По този начин се търсят грешки в дълги сметки. Забележителна подробност е, че двоичното търсене може да се прилага в задачи като тази, в които масивът не е сортиран.

- 3) Търсим индекса на елемент на сортиран списък по дадена стойност на елемента.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.

Обосновка: Кандидати са само последователното и двоичното търсене. Двоичното търсене е по-бързото от двете, но то изисква пряк достъп, тоест масив, а не списък от входни данни. Затова последователното търсене е единственият приложим метод от петте предложени.

- 4) Търсим индекса на най-малък елемент в несортиран масив.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.

Обосновка: Освен чрез последователно търсене, задачата може по принцип да се реши още чрез бързо търсене и чрез алгоритъма PICK. Времето и на трите метода е линейно, но все пак PICK и бързото търсене са много по-сложни от последователното търсене, затова константният множител в неговата времева сложност е по-малък от техните и то е най-подходящо в тази задача.

- 5) Търсим елемент на несортиран масив, по-голям от 25% от елементите, за най-малко време, усреднено по всевъзможните входни данни.
а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.
г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.

Обосновка: Тази задача може да се реши чрез бързо търсене и чрез алгоритъма PICK. Бързото търсене е по-подходящо, тъй като искаме да оптимизираме средната сложност.