С О Р Т И Р А Н Е И Т Ъ Р С Е Н Е
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 вмъкване. в) Пирамидално сортиране.
   г) Метод на бройните системи. д) Бързо сортиране. е) Сортиране чрез сливане.

2) Да се сортира устойчиво и възможно най-бързо масив от дължини на отсечки.
   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 най-бързо масив от бройките научни публикации на няколко изследователи.
   а) Сортиране чрез сливане. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
   г) Метод на мехурчето. д) Бързо сортиране. е) Метод на бройните системи.

4) Да се сортира на място 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 сортират на място физически обекти (например щайги) с най-малък брой движения.
   а) Сортиране чрез броене. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
   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 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.

2) Търсим индекса на елемент на несортиран 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.

3) Търсим индекса на елемент на сортиран масив 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) В несортиран масив търсим най-голям брой елементи със сбор, по-малък от дадено число.
   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 при най-лоши 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 вмъкване. в) Пирамидално сортиране.
г) Метод на бройните системи. д) Бързо сортиране. е) Сортиране чрез сливане.

Обосновка: Сортирането чрез броене и методът на бройните системи имат специфични изисквания към входа, за които не е казано, че са изпълнени. Пирамидалното и бързото сортиране сравняват далечни елементи, тоест работят върху масиви, а не върху списъци. Сортирането чрез сливане копира данните, тоест не сортира на място. Следователно само сортирането чрез вмъкване е подходящо в тази задача (от предложените отговори).

2) Да се сортира устойчиво и възможно най-бързо масив от дължини на отсечки.

а) Сортиране чрез броене. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Бързо сортиране. е) Метод на бройните системи.

Обосновка: Сортирането чрез броене и методът на бройните системи са неприложими към дробни числа (например дължини на отсечки). От другите четири метода най-бързи са пирамидалното и бързото сортиране (време: $n \log n$); останалите два са квадратични. Но пирамидалното сортиране е неустойчиво. Като отговор остава бързото сортиране.

3) Да се сортира най-бързо масив от бройките научни публикации на няколко изследователи.

а) Сортиране чрез сливане. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Бързо сортиране. е) Метод на бройните системи.

Обосновка: Броят на научните публикации на един автор е сравнително малко цяло число, рядко по-голямо от сто. Затова можем да използваме метода на бройните системи, който работи в линейно време. Другите алгоритми са по-бавни.

4) Да се сортира на място 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 при дробни числа (тегла на предмети). От другите алгоритми най-бързи са пирамидалното сортиране и сортирането чрез сливане. Обаче сортирането чрез сливане не работи на място. Остава пирамидалното сортиране.

5) Да се сортират на място физически обекти (например щайги) с най-малък брой движения.

а) Сортиране чрез броене. б) Сортиране чрез вмъкване. в) Пирамидално сортиране.
г) Метод на мехурчето. д) Бързо сортиране. е) Сортиране чрез пряк избор.

Обосновка: Сортирането чрез пряк избор извършва най-малко разместванията. Този факт е известен от теорията.

Задача 2. Верните отговори са подчертани.

1) 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 последователното и двоичното търсене (останалите алгоритми решават други задачи). Тук двоичното търсене не работи, нищо че данните са сортирани. Причината е, че двоичното търсене разчита на достъп по индекс, т.е. работи върху масиви, а не върху списъци. Така остава само последователното търсене.

2) Търсим индекса на елемент на несортиран 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: Същата като на предишния въпрос. Няма значение дали списъкът е сортиран. Важното е, че имаме списък, а не масив.

3) Търсим индекса на елемент на сортиран масив 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: Този път имаме масив, а не списък, затова прилагаме двоично търсене. То е по-бързо от последователното търсене. (Останалите алгоритми са за други задачи.)

4) В несортиран масив търсим най-голям брой елементи със сбор, по-малък от дадено число.

а) Последователно търсене. б) Двоично търсене. в) Бързо търсене.

г) PICK (самостоятелно). д) PICK в съчетание с двоично търсене.

Обосновка: Това съчетание се използва точно за този вид задачи.

5) Търсим елемент на несортиран масив, по-голям от 25% от елементите, 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: Приложими в тази задача са алгоритъмът PICK и бързото търсене. Обаче бързото търсене е по-добро от алгоритъма PICK само в смисъл на средно време, докато при лоши входни данни е обратното. Затова в този случай PICK е по-добрият избор.