1. *Обяснете каква е разликата между средна и амортизирана сложност на алгоритъм. Дайте пример.*

Средната сложност е средното количество ресурс (памет/време) нужно за еднократно изпълняване на алгоритъма.

Амортизираната сложност е също средното количество ресурс, но този път за няколко последователни изпълнения на алгоритъма (т.е. тук се взима в предвид, че някои операции може да използват повече ресурси от други, но за достатъчно много последователни изпълнения клоним към амортизираната сложност).

Пример:

Операция добавяне на елементи в масив, който при напълване удвоява размера си.

Времева сложност:

O(1) при добавяне без оразмеряване.

O(N) при добавяне с оразмеряване.

Средно O(N)

Амортизирано O(1)

1. *Нека е даден****двусвързан****списък, който съдържа N елемента. Считаме, че представянето е такова, че разполагаме с указатели към първата и последната кутия в списъка.*

*Каква ще бъде сложността на всяка от дадените по-долу операции със списъка?*

Сливане на два сортирани списъка: O(N)

1. *Направете сравнение между масив и едносвързан списък:*
2. *Обяснете накратко какво представлява всяка от двете структури;*
3. *За всяка от тях посочете кои са нейните предимства;*
4. *За всяка от тях посочете кои са нейните недостатъци.*

Масивът е последователно наредени в паметта данни от еднакъв тип.

Предимства:

1. Бързо достъпване на произволен елемент (индексиране).

2. Бързо изтриване на последния елемент.

3. Има много добро 'data locality' и заради това обхождането му подред е много бързо.

Недостатъци:

1. Когато се напълни трябва да се оразмери, което е бавна операция изискваща допълнително памет.

2. Изтриването на произволен елемент е бавно О(N) (трябва задните да се преместят).

3. Ако не е сортиран търсенето е О(N).

4. Добавянето е бавно О(N).

Едносвързаният списък е 'верига' от еднотипни данни. Заедно с данните има и препратка (указател) към следващия елемент във веригата.

Предимства:

1. Бързо достъпване и триене на първия елемент във веригата.

2. Не е нужна тежката операция преозармеряване както при масива.

3. Обикновено не се заделя допълнителна памет като при масива, за да се избегне преоразмеряване.

4. Триенето е О(1) (ако имаме препратка към предходния елемент).

 Недостатъци:

1. Лошо data locality в общия случай.

2. Няма индексиране. Достъп до произволен елемент О(N).

3. Във всеки елемент освен за данните трябва да има памет и за препратката.

1. *Нека са дадени четирите числа 1, 5, 10 и 100. По колко начина тези числа могат да бъдат наредени в двоично дърво, така че то да бъде двоично дърво за търсене (binary search tree)?*

Отг: 14

1. *Обяснете накратко какво означава за едно двоично наредено дърво да бъде "изродено".*

*Обяснете как израждането влияе върху операциите с дървото.*

Изродено дърво е дърво с дълбочина от порядъка на броя на елементите му (в най-лошият случай дървото става на практика свързан списък: верига от елементи без разклонения).

Колкото по-изродено е едно двоично наредено дърво, толкова повече операциите с него клонят към времева сложност O(N) вместо O(logN).