

СДА Обобщение

Лекция 14 по СДА, Софтуерно Инженерство
Зимен семестър 2019-2020г
д-р Милен Чечев

План за лекцията

- Преглед на темите разгледани в курса
- Обобщение
- Съвети/Подготовка за интервю/стаж

Конспект (от първата лекция)

1. Оценка на сложност за алгоритми.
2. Алгоритми за сортиране (Bubble, Selection, Insertion, Merge, Quick, Counting)
3. Алгоритми за търсене (Linear, Binary, Ternary Search)
4. Свързан списък. Специфични особености. Реализации.
5. Стек и опашка. Shunting-yard algorithm.
6. Дървета.

- Двоично дърво.
- Балансирани дървета. ~~2 3 4, Red Black, AVL, Treap~~
- Heap. Priority Queue
- ~~B-tree~~

7. Граф

- Какво е граф и основни представяния и имплементации
- Алгоритми за търсене в дълбочина и широчина в граф
- Топологично сортиране
- Цикъл в граф. Ойлеров и Хамилтонов цикъл в граф.
- Алгоритми за Минимално покриващо дърво (Prim, Kruskal)
- Търсене на най-кратък път в граф. Алгоритъм на Dijkstra.

8.8 Хеширане

- Какво е хеш функция. Свойства на хеш функциите. Какво е колизия. Видове хеш функции. Как да се справяме с колизии. Приложения на хеширането.
- Структури от данни използващи хеш функции: Хеш Таблицы и Хеш Мапове.
- ~~Bloom Filters - идея, основни свойства и приложения в реалния свят.~~

СДА Конспект(обобщение)

1. Сложност на алгоритми. Оценка на сложност на алгоритъм. Big O нотацията. NP Complete проблеми.
2. Алгоритми за Сортиране. Бавни алгоритми: Метод на мехурчето, на пряка селекция, сортиране с вмъкване, Бързи алгоритми: сортиране със сливане, бързо сортиране, Алгоритми със ограничения: сортиране с броене
3. Алгоритми за търсене - линейно, двоично
4. Списък, Стек, Опашка - реализация, основни операции и сложности
5. Дървета, Двоично дърво за търсене, Балансирани дървета, Пирамида
6. Хештаблица/хешсет
7. Граф. Обхождане в дълбочина, ширина, Минимално покриващо дърво, алгоритъм на Дейкстра

Какво да уча за изпита

Всичко, включено в курса!

Как да се подготвя за изпита?

1. Разгледайте отново всички презентации, докато ги научите
2. Прегледайте всички задачи от контролни
3. Прегледайте всички задачи от домашни

Трябва ли да мога да имплементирам различните алгоритми/структури или само да ги ползвам?

- Трябва да разбирате как работят.
- Да можете да ги използвате
- Да може да ги разширите/промените
- За последното е нужно да ги разбирате и да можете да ги имплементирате.

Защо да уча толкова много за един предмет/изпит?

- Защото материала от СДА е основен за изпитване и на интервюта!

Какво ще ме питат на интервю в Гугъл?

СДА! Както и в повечето големи технологични компании.

1. Какви структури от данни знаеш?
2. Какво е Свързан списък?
3. Какво е хешмап? Хеш функция? Колизии и справяне с тях?
4. Какви са сложностите на операциите в различни структури от данни?
5. Реализирай тази задача за дърво/граф или
6. Реализирай тази задача, която ползва хеш мап/сет и свързан списък
7. Каква е сложността на написаното решение?
8. Може ли да се подобри?
9. Каква е сложността на подобреното решение

Примерна задача от Гугъл интервю

<https://www.hackerrank.com/sda-test>

Решение на задачата

Проблем: Да определим дали в масив имаме 3 числа (a,b,c) такива че $a+b=c$

Стандартно решение:

```
for(int i = 0 ; i < arr.length;i++)
```

```
    for(int j = 0; j < arr.length; j++)
```

```
        for(int k = 0 ; k < arr.length; k++)
```

```
            if(arr[i]+arr[j] = arr[k] && i!=j && j!=k && k!=i)
```

```
                return "true"
```

```
return "false"
```

$O(N^3)$

Бързо решение

```
// броим и проверяваме
HashSet<Integer> set = new HashSet<>()
for(int i = 0 ; i < arr.length ; i++){
    for( int j = i+1; j < arr.length; j++){
        set.add(count[arr[i]+arr[j]]);
    }
}
for(int i = 0 ; i < arr.length; i++){
    if(set.contains(arr[i])){
        return true
    }
}
return false;
```

$O(N^2)$

Съвети за интервю

- Преговорете си материала от структури от данни и алгоритми
- Решете на лист/бяла дъска 1-2 задачи за да сте по-уверени за coding задача на интервю
- Когато решавате задача, първо намерете baseline решение, споделете го с интервюиращият, определете сложността му и евентуално го реализирайте(освен ако веднага не измислите решение с по-малка сложност)
- Внимавайте в детайлите, питайте въпроси за ограничения в данните и гранични случаи.

Кариерни съвети

- В България или в Чужбина?
 - Трябва да сте толкова добри, че да работите и живеете там където искате. Ако това място е в Австралия - добре.
 - България е едно от най-добрите места за живот като програмист!
 - Има супер иновативни и интересни компании за работа в България!

Кариерни съвети

- Направете си LinkedIn профил и CV на английски в PDF формат, което ви показва максимално добре нещата които сте правили до момента - Професионален опит, Езици за програмиране, проекти, github, hackerrank, kaggle, coursera
- Добавете и обобщение с това какво сте правили до момента и с какво искате да се занимавате.
- Започнете да трупате професионален опит
 - За да комбинирате с университета - има и опции за почасова работа или платен стаж.
 - Ако се чудите къде да почнете изберете мястото на което се влиза по-трудно
 - Очаквайте скоро съобщение за платени стажове в фирмата в която работя:

Интересен факт: CEO-то на Окадо технолоджи е бил стажант в компанията в началото на кариерата си: [LinkedIn](#)



Това е всичко за днес.

Какво следва:

- 24.01.2018 13:00 изпит в 210 и 130 зала на химическият.
 - Носете си лаптопи и студентските книжки!