Алгоритмы и структуры данных

Лекция 3*. Задачи на построение алгоритмов.

(c) Глухих Михаил Игоревич, glukhikh@mail.ru

Максимальный подмассив

- Есть (известен) массив цен на акции Price[]
- Требуется выбрать два дня i, j, такие, что
  - j > i (!!!)
  - Price[j] – Price[i] → MAX
- То есть ВНАЧАЛЕ купить дёшево, ПОТОМ продать дорого
- Примеры
  - 100, 113, 110, 85, 105, 102, 86, 63, 81, 101, 94, 106, 101, 79, 94
  - 10, 11, 7, 10, 6

Максимальный подмассив

- Есть (известен) массив цен на акции Price[]
- Требуется выбрать два дня i, j, такие, что
  - j > i (!!!)
  - Price[j] – Price[i] → MAX
- То есть ВНАЧАЛЕ купить дёшево, ПОТОМ продать дорого
- Примеры
  - 100, 113, 110, 85, 105, 102, 86, **63**, 81, 101, 94, **106**, 101, 79, 94
  - 10, 11, **7**, **10**, 6

Вариация

- Берём массив цен Price[]
- Формируем массив изменений Delta[] :
  Delta[i] = Price[i+1] – Price[i]
- Требуется найти подмассив Delta, такой, что сумма его элементов максимальна
  - МАКСИМАЛЬНЫЙ ПОДМАССИВ

Варианты решения

- В лоб
- Разделяй и властвуй
- Рекуррентный

Варианты решения

- В лоб
  - Просто перебираем все варианты: $O(N^2)$
- Разделяй и властвуй
  - Разбиваем массив дельт на две половины
  - В каждой ищем максимальный подмассив
  - Сравниваем его с тем, который проходит через обе половины

Варианты решения

- Разделяй и властвуй
  - Разбиваем массив дельт на две половины
  - В каждой ищем максимальный подмассив
  - Сравниваем его с тем, который проходит через обе половины
- Рекуррентный
  - Ищем максимальный подмассив для массива размером J-1
  - Далее одно из двух
    - Либо он максимален и для размера J
    - Либо для размера J максимален подмассив, включающий последний элемент

Задача умножения матриц

- Упрощённый вариант: квадратные матрицы размером M x M, причём M является степенью двойки
- Вариант “в лоб”
  - Вычисление каждого элемента произведения требует M умножений и M-1 сложений
  - Элементов имеется M^2
  - Общая трудоёмкость $O(M^3)$
- Рекурсивный вариант
  - Делим каждую матрицу на четыре подматрицы
    - A_{11} A_{12} B_{11} B_{12}
    - A_{21} A_{22} B_{21} B_{22}
  - Перемножаем подматрицы отдельно, потом сливаем вместе
  - $S(M) = 8S(M/2) + O(M^2) \implies S(M) = O(M^3)$

Метод Штрассена

- Идея: уменьшить число перемножений подматриц с 8 до 7, сведя рекуррентное соотношение к следующему
  - S(M) = 7S(M/2) + O(M2) ==> S(M) = O($M^{2.81}$)
- С этой целью из 8 подматриц A_{ij} / B_{ij}
  формируется ещё 10 матриц $S_k = A_{ij} +/- B_{mn}$,
  вычисляется 7 произведений,
  суммируются результаты для формирования A x B
- Подробности см. книгу Кормена

10

Наибольшая общая подстрока

- Даны две строки (first, second)
- Найти их наибольшую общую подстроку
- Решение в лоб?

Наибольшая общая подстрока

- Даны две строки (first, second)
- Найти их наибольшую общую подстроку
- Решение в лоб?

- Тройной цикл: по номеру символа двух строк и общей подстроки

Наибольшая общая подстрока

	О	Б	С	Е	Р	В	А	Т	О	Р	И	Я
К	0	0	0	0	0	0	0	0	0	0	0	0
О	**1**	0	0	0	0	0	0	0	**1**	0	0	0
Н	0	0	0	0	0	0	0	0	0	0	0	0
С	0	0	**1**	0	0	0	0	0	0	0	0	0
Е	0	0	0	**2**	0	0	0	0	0	0	0	0
Р	0	0	0	0	**3**	0	0	0	0	**1**	0	0
В	0	0	0	0	0	**4**	0	0	0	0	0	0
А	0	0	0	0	0	0	**5**	0	0	0	0	0
Т	0	0	0	0	0	0	0	**6**	0	0	0	0
О	**1**	0	0	0	0	0	0	0	**7**	0	0	0
Р	0	0	0	0	**1**	0	0	0	0	**8**	0	0
Ы	0	0	0	0	0	0	0	0	0	0	0	0

Итоги

- Рассмотрели
  - Рекуррентный и рекурсивный (разделяй и властвуй) подходы
  - Оценки эффективности
  - Наилучший / наихудший случай
  - Примеры применения на практике
- Далее
  - Алгоритмы сортировки