Домашній Валентин Олексійович
ІП-93

Білет № 11

1. Метод декомпозиції — це рекурсивний підхід, коли завдання, яке є початковим, розділяють на менші задачі (підзадачі), розв'язання яких певним способом обробляються для того, щоб отримати рішення всього завдання. Підзавдання можуть знову розділяти на підзавдання. Таким чином ми доходимо до найменшого розміру завдання. Ця парадигма має назву „розділяй та володарюй" та має три етапи: – Поділ завдання на підзавдання. – Рекурсивне розв'язування підзавдань. До того моменту, коли підзадачі будуть достатньо малими, щоб безпосередньо розв'язати. – Поєднання розв'язання початкового завдання з розв'язків підзавдань. Почнемо розбирати цю парадигму з „merge sort" тобто алгоритму сортування методом злиття. За допомогою „розділяй та володарюй" опишемо цей метод: – Поділ: низка послідовностей розбивається на дві менші низки (перша низка складається з x елементів, а після поділу маємо дві нові з x/2 елементами). – Розв'язання за допомогою рекурсії: нам треба відсортувати обидві створені послідовності за допомогою методом злиття тільки тоді, коли їх розмір перевищує один. – Об'єднання: об'єднання двох відсортованих низок послідовностей, щоб отримати фінальний результат.
Підрахунок інверсій:
Для кращого розуміння наведу приклад: маємо масив чисел X, який дорівнює <2, 3, 8, 6, 1>. Тут пара індексів буде (1,5) (саме ця пара є інверсією). Загальна к-сть інверсій = 5 ((3, 5), (2, 5), (1,5), (3,4), (4,5)). Якщо ж ми маємо відсортований масив, то к-сть інверсій = 0; якщо масив відсортований у зворотньому напрямку, то застосуємо формулу $\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$. Таким чином к-сть інверсій буде максимальна і для реального масиву = 10.

2. Алгоритм Прима — жадібний алгоритм для побудови мінімального кістякового дерева зваженого зв'язного неорієнтованого графа. Цей алгоритм у 1930 році розробив чеський математик Войтех Ярник, а потім вчені Роберт Прим у 1957 та Едсгер Дейкстра в 1959 знову відкрили та перевидали його. Тому цей алгоритм має різні назви: алгоритм DJP, алгоритм Прима-Дейкстри, алгоритм Ярника та алгоритм Прима-Ярника. Цей алгоритм доволі швидкий для розріджених графів, але в той же час програє іншим, більш складним алгоритмам. Уже для графів, які є достатньо щільними алгоритм Прима можна змінити і таким чином його швидкість стане O(n). Побудова починається з дерева, що має в собі одну (будь-яку) вершину. Під час роботи алгоритму дерево стає все більшим, поки не буде мати в собі всі вершини початкового графа. На кожному кроці алгоритму до поточного дерева додається найменше ребро з тих, що з'єднують вершину з побудованого дерева і вершину, що не належить дереву.
Алгоритм: 1) Ребра сортують за зростанням ваги. 2) Додають найменше ребро в дерево 3) Зі списку ребер із найменшою вагою вибирають таке ребро, щоб одна з його вершин належала дереву, а інша — ні. 4) Це ребро додають у дерево і знову переходять до кроку 3. 5) Робота закінчується, коли всі вершини будуть у дереві.

3. Рекурентне співвідношення — рівняння або нерівність, яка описує функцію із використанням самої себе, але тільки з меншими аргументами. При розгляді рекурентних співвідношень зазвичай є два спрощення. По-перше, приймається, що час роботи алгоритму $T(n)$ визначається лише для цілочислових n, оскільки в більшості алгоритмів к-сть вхідних елементів виражається цілим числом. По-друге, оскільки час роботи алгоритму з вхідними даними фіксованого розміру виражається константою, то в рекурентних співвідношеннях для достатньо малих n зазвичай справедливо положення $T(n) = \Theta(1)$. Тому для зручності граничні умови рекурентних співвідношень, як правило, відкидаються та приймається, що для малих n час роботи алгоритму є $T(n)$ константою. Метод підстановки, який застосовується для рішення рекурентних співвідношень, складається з двох етапів: 1) робиться припущення про вигляд розв'язку; 2) за допомогою методу математичної індукції визначаються константи та доводиться, що рішення вірне. Назва методу пояснює, що припущене рішення підставляється у рекурентне рівняння. Цей метод достатньо потужний, але може застосовуватись лише коли легко можна висунути припущення щодо вигляду розв'язку.

На жаль, не існує єдиного рецепту, як вгадати розв'язок рекурентного співвідношення. Для цього потрібен досвід, вдача та творче мислення. Втім існують певні евристичні прийоми, які можуть допомогти зробити правильну здогадку. Якщо рекурентне співвідношення подібне до того, що ми щойно розглянули, то можна припустити, що розв'язки цих співвідношень будуть подібними. Наприклад, розглянемо рек. спів.: $T(n)=2T(\lfloor n/2 \rfloor+17)+n$, яке виглядає більш складним, адже в аргументі функції T правої частини додається доданок 17. Але інтуїтивно зрозуміло, що цей доданок не може вплинути на асимптотичну поведінку рішення. При достатньо великому n різниця між $T(\lfloor n/2 \rfloor)$ та $T(\lfloor n/2 \rfloor+17)$ стає несуттєвою: обидва ці числа приблизно рівні половині числа n. Відповідно, можна припустити, що $T(n) = O(n \lg n)$ і в цьому випадку, та за допомогою методу підстановки перевірити це припущення. Інший спосіб знайти спосіб шукати розв'язок — з початку грубу оцінку верхньої та нижньої межі, а потім зменшити невизначеність до мінімуму. Наприклад, в рекурентному співвідношенні в якості початкової нижньої межі можна було б обрати $T(n) = \Omega(n)$, оскільки в ньому присутній доданок n; можна також показати, що грубою верхньою межею буде $T(n) = O(n^2)$. Далі верхня межа поступово знижується, а нижня — піднімається доки не буде отримана правильна асимптотична поведінка рішення $T(n) = O(n \lg n)$