Динамическое программирование I

1. В графе может быть несколько кратчайших путей между какими-то вершинами. Постройте линейный по времени алгоритм, находящий в заданном неориентированном графе с единичными весами количество различных кратчайших путей между заданными двумя вершинами.

Опредедение. Пусть G — связный неориентированный граф, а $d_{u,v}$ — кратчайшее расстояние между вершинами u и v. *Диаметром* графа G называют число $\max\limits_{u,v} d_{u,v}$ (максимальную из длин кратчайших простых путей между каждой парой вершин графа).

2. Постройте алгоритм, который находит диаметр дерева за $O(|V|)$.

3. Рассмотрим следующую игру. На доске нарисовано n палочек. Два игрока по очереди зачёркивают от одной до трёх палочек. Проигрывает тот, кто зачеркнул последнюю палочку.

1. Кто выигрывает при $n = 20$? (Считая, что соперник не ошибается.)

2. Кто выигрывает при произвольном n? Постройте алгоритм, который решает задачу **а)** динамическим программированием; **б)** жадным алгоритмом.

4. Фирма производит программное обеспечение для банкоматов разных стран мира. Банкомату нужно выдавать запрашиваемую клиентом сумму минимальным количеством купюр.

1. Если у банкомата есть купюры номиналом 1, 2, 5, 10, 20, 50, а сумма — 71, то набор банкнот будет 50+20+1. Постройте жадный алгоритм, который будет решать задачу для данного набора купюр и произвольной суммы, которая является входом задачи.

2. Постройте алгоритм, который решает задачу, в случае когда на вход помимо суммы подаются и номиналы банкнот. Является ли он полиномиальным от длины входа?

5. На столе лежат в ряд n банковских карт, на счету которых находится $v_1, v_2, \ldots v_n$ долларов. Два игрока по очереди берут карты, причём можно брать только крайнюю карту (с любой стороны), до тех пор, пока не кончатся все карты. Каждый из игроков, заинтересован в максимальной суммарной стоимости взятых им карт. Считайте, что n чётно.

1. Покажите, что жадная стратегия (брать ту из карт, которая дороже) не оптимальна: уже первый ход по этому правилу может помешать достичь оптимума.

2. Постройте алгоритм отыскания оптимальной стратегии для первого игрока за время $O(n^2)$. Получив на вход $v_1, \ldots, v_n$, алгоритм должен произвести препроцессинговые вычисления за $O(n^2)$, а вычисление каждого следующего хода должно выполняться за $O(1)$ шагов (с использованием сохранённой информации).

3. Представьте теперь, что задача игрока не максимизировать стоимость карт, а просто набрать стоимость больше, чем у соперника. Постройте жадное решение для этой задачи.