Домашнее Задание по алгоритмам №12

Павливский Сергей Алексеевич , 873

28.04.2019

Задача 1.

Как модифицировать алгоритм Флойда-Уоршелла, чтобы он находил не только длины кратчайших путей между всеми парами вершин, но и сами пути?

Решение

Дополнительно заведем массив A, где для каждой вершины будем хранить путь до нее . При записывании в вершину a кратчайшего пути будем также записывать в соответсвующий ей элемент массива A значение вершины A , соответствующей вершине b , из которой мы пришли в a , в конце которого приписываем a . Итого после окончания работы алгоритма каждой вершине будет соответствовать последовательность вершин , которая и будет кратчайшим путем до этой вершины из исходной .

Задача 2.

Как используя выходные данные алгоритма Флойда-Уоршелла проверить, что в графе есть цикл отрицательного веса?

Решение

Так как алгоритм Флойда последовательно релаксирует расстояния между всеми парами вершин (i,j), в том числе и теми, у которых i=j, а начальное расстояние между парой вершин (i,i) равно нулю, то релаксация может произойти только при наличии вершины k такой, что d[i][k]+d[k][i]<0, что эквивалентно наличию отрицательного цикла, проходящего через вершину i.

Значит при наличии цикла отрицательного веса в матрице появятся отрицательные числа на главной диагонали.

То есть проходимся по элементам выходной матрицы A[i][i] , где i от 0 до |V| - 1 , и если хоть один из встреченных элементов < 0 , то в графе есть цикл отрицательного веса .

Задача 3.

В ориентированном взвешенном графе есть ровно одно ребро (u → v) с отрицательным весом. Описать эффективный алгоритм поиска кратчайшего пути между заданной парой вершин (a, b) — вход задачи: матрица весов и вершины a и b.

Решение

За $|V|^2$ проедмся по всем ребрам и найдем ребро отрицательного веса . Дальше дальше запускаем поиск в ширину из a не учитывая отрицательное ребро , потом запускаем поиск в ширину из v также не учитывая отрицательное ребро . Дальше min(d[a][b], d[a][u] + d[u][v] + d[v][b]) и будет ответом .

Ассимптотика : 2 * BFS + $|V|^2$ = O($|V|^2$) .

Задача 4.

В Главе 2 [ДПВ] (раздел 2.5) приведён алгоритм Штрассена для умножения матриц сложностью O($n^{log 7_2}$).

Решение

Задача 5.

Предложите $O(|V| + |E|)$ алгоритм, который находит центр дерева (вершину, максимальное расстояние от которой до всех остальных минимально). Докажите его корректность и оцените асимптотику.

Решение

Будем удалять последовательно удалять листья (все листы заносятся в очередь в произвольном порядке , когда удаляем лист его родитель заносится в конец очереди) . Так делаем до тех пор , пока не останется 1 или 2 вершины . Та вершина , на которой процесс остановится и будет центром дерева .

Ассимптотика : мы проходимся по $O(|V|)$ вершинам , также мы обрабатываем $O(|E|)$ ребер (когда заносим родителей в очередь) . Суммарная ассимптотика : $O(|V| + |E|)$, что и требовалось .

Задача 6.

Решение

Задача 7.

Даны две последовательности $x[1] \dots x[n]$ и $y[1] \dots y[m]$ целых чисел. Постройте алгоритм, который находит максимальную длину последовательности, являющейся подпоследовательностью обеих последовательностей. Сложность алгоритма $O(nm)$.

Решение

Пройдемся последовательно по всем элементам последовательности x . Также дополнительно для каждого элемента x будем хранить переменную , которая будет считать максимальную длину подпоследовательности на данный момент , а также переменную , хранящую номер элемента в массиве y , если был найден совпавший . Для каждого элемента пройдемся по элементам массива y в поисках равного ему элемента . Если мы находим совпадающий элемент , то мы сравниваем данный элемент x с предыдущим и в случае , если рассматриваемый сейчас больше предыдущего и номер совпавшего элемента в массиве y больше номера совпавшего элемента в массиве y для предыдущего , то x.count[i] = x.count[i - 1] + 1 , иначе x.count[i] = 1 . Ответом будет max(x.count[i]) .

Ассимптотика : так как для n эл-ов проходим m эл-ов , то O(m · n)