

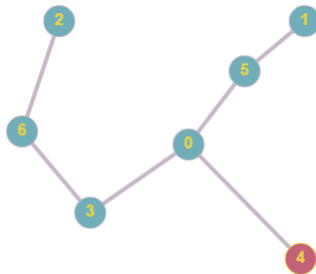
Problem 12

Постановка задачи полностью совпадает с задачей *Euclidian TSP* (Евклидова задача Комми-Вояжёра), поэтому буду говорить в терминалогии графа в евклидовой плоскости. Вот несколько возможных алгоритмов:

I) Жадный алгоритм Nearest Neighbour (NN). На каждой итерации жадного алгоритма выбираем ближайшую вершину к последней выбранной и добавляет её в путь. Оценка эффективности: полученный путь не может быть более, чем вдвое хуже оптимального возможного.

II) Алгоритм состоит из нескольких пунктов:

- а) Ищем минимальное остовное дерево (например алгоритмом Краскала);
- б) Выбираем начальную вершину, проводим от неё поиск в глубину;
- в) Убираем из полученной последовательности все повторяющиеся вершины



В примере на рисунке, после б) путь $[0, 3, 6, 2, 6, 3, 0, 5, 1, 5, 0, 4]$, тогда после пункта в) он станет $[0, 3, 6, 2, 5, 1, 4]$

Оценка эффективности: полученный путь не может быть более, чем вдвое хуже оптимального возможного.

- III) Алгоритм Кристофидеса. Схож с прошлым пунктом, но после нахождения остовного дерева
- б) Для всех вершин нечётной степени строится идеальное паросочетание, после чего рёбра из него добавляются в дерево
 - в) В полученном графе ищется эйлеров цикл (теперь он точно есть, так как все вершины имеют чётную степень), который и является ответом. Оценка в данном случае: полученный путь не может быть более, чем в полтора раза хуже оптимального возможного.

В общем случае из оценки неизвестно, какой из алгоритмов найдёт лучший ответ, поэтому стоит запустить каждый и найти лучший.

Хорошей эвристикой для всех алгоритмов является удаление всех пересекающихся рёбер в полученном цикле (если в цикле рёбра AB и CD пересекаются, то вместо них можно взять рёбра AD и BC , тогда полученный граф всё равно будет циклом, но длина цикла будет меньше (это тривиальный факт)).

В приложенном файле написал алгоритм поиска, основанный на жадном алгоритме NN .