Разбор задачи «Поиск корабля»

В условии задачи описан неориентированный невзвешенный граф. Обозначим через d_i длину кратчайшего пути от вершины s до вершины i. Значения d_i можно найти, запустив в графе поиск в ширину.

Обозначим через T_i множество всех вершин, лежащих на хотя бы одном кратчайшем пути из вершины s в вершину i. Найдем множества T_i для всех вершин графа. Для этого упорядочим вершины по возрастанию значения d_i, и в этом порядке будем строить искомые множества для вершин. Заметим, что если какой-то кратчайший путь из вершины s заканчивается в вершине v, то предпоследняя вершина этого пути может быть расположена в вершине u тогда и только тогда, когда $d_v = d_u + 1$, а также вершины u и v соединены ребром. Таким образом, множество T_v — это объединение множеств T_u для всех подходящих вершин u. Также в множество T_v нужно не забыть добавить саму вершину v.

Для того, чтобы быстро построить множества T_i, воспользуемся битовым сжатием: его можно реализовать самостоятельно, либо с помощью структуры данных bitset. Эта структура данных позволяет хранить множества чисел от 1 до n и производить операцию объединения за время $O(\frac{n}{w})$, где w — машинное слово, равное 32 или 64, в зависимости от параметров системы.

Для того, чтобы ответить на запрос, необходимо выяснить, какие вершины лежат на кратчайшем пути от s до v, причем расстояние до самих этих вершин от s равно k. Для того, чтобы получить множество всех таких вершин, возьмем множество вершин, которые лежат на кратчайших путях из s в v, и удалим из него все вершины, кроме тех, для которых $d_i = k$. Заметим, что так как мы упорядочили вершины в порядке возрастания величины d_i, искомые вершины будут образовывать непрерывный подотрезок, а значит, необходимые операции удаления с использованием битового сжатия можно сделать за время $O(\frac{n}{w})$, используя операции битового сдвига. После того, как в множестве остались лишь подходящие вершины, необходимо вывести ответ: если множество пусто, то ответ 0, если в нем более одного элемента, то ответ -1, а в противном случае ответ — это вершина, которая содержится в найденном нами множестве. Для проверки всех этих условий за время $O(\frac{n}{w})$ также можно использовать битовое сжатие.

Асимптотика времени работы программы, а также памяти, используемой программой, составляет $O(\frac{n(m+q)}{w})$.