Бабушкин А.

1 Обязательные задачи

1.
   a) Поделим на k, решим 1-2, домножим ответ на k.
   b) У нас будут очереди, и в i-ой очереди лежат вершины с расстоянием из [ik, (i + 1)k). Дальше как в 1-2.

2. Давайте сделаем m/n-ичную кучу.

3. Запихаем все вершины из A в очередь, проставив им расстояние 0, и запустим bfs. Так мы найдём кратчайшее расстояние, а количество путей потом считаем динамикой на ДАГе, оставив только те рёбра a-b, для которых d[a] + 1 = d[b].

4. Запустим Флойда. Ответ на запрос – ребро s-t веса w, хорошее, если d[a][b] = d[a][s] + w + d[t][b] или d[a][b] = d[a][t] + w + d[s][b].

5. Запустим Дейкстру только на v и вершинах из A. Оставим только полезные рёбра, посчитаем на ДАГе сколько есть кратчайших путей из v в каждую вершину. Затем сделаем то же самое на всех вершинах. Если у вершины количество кратчайших путей в первом случае и во втором не равно, то она хорошая.

6. Назовём состоянием пару вершин (s, t), то есть первый стоит в s, второй – в t. Нам надо попасть из (v, u) в (u, v). Запустим bfs по состояниям. Почему работает nm? Для ребра a-b мы рассмотрим его только в тех состояниях, где одна из вершин или a, или b, а вторая вершина может быть любой, то есть в O(n) состояниях. Успех.

2 Дополнительные задачи

1. Запустим от s Дейкстру, которая минимизирует два наибольших ребра на пути, запустим такую же по обратным рёбрам из t.
   Теперь переберём вершину v. Пусть первая Дейкстра дала пару (a, b), а вторая – (c, d). Тогда если b <= c и d <= a, то обновим ответ значением a + c. Почему работает? Рассмотрим оптимальный путь. На нём точно есть вершина, которая лежит между двух самых тяжёлых рёбер. Тогда в такой вершине будет правильный ответ.