Домашняя работа № 2

Автор: Минеева Екатерина

Задача B2. Бензин и деньги

Для решения задачи воспользуемся динамикой. Будем хранить в массивы:

- $wayLen[i]$ – длина кратчайшего (по затратности бензина) пути из $start$ (начальной вершины) в v_i, среди тех, на которые в принципе хватает бензина.

- $maxPetrol[i]$ – максимальное количество бензина, которое мы могли бы залить в бак, проехав по пути в v_i, длина которого записана в $wayLen[i]$.

- $prevVertex[i]$ – предпоследняя вершина в пути, длина которого записана в $wayLen[i]$. Этот массив понадобится для восстановления ответа.

Пусть также у нас есть массив: $petrol[i]$ – максимальное количество бензина, которое могут заправить в городе v_i. Это – часть входных данных.

Начальные условия:

- $wayLen[start] = 0$ – Если начальная вершина, совпадает с конечной, то ехать никуда не нужно, а значит, длина пути - 0. Во всех остальных случаях – бесконечность.

- $maxPetrol[start] = petrol[start]$ – максимальное количество бенизна, которое можно заправить в городе $strat$, больше бензин брать неоткуда. Для остальных веришин – пока 0.

- $prevVertex[start] = -1$ – поскольку весь путь состоит из одной вершины, то предпоследняя вершина пути не определена. В наших обозначениях это будет -1. В остальных случая тоже -1, поскольку ни один путь еще не построен.

Пересчет:

Для пересчета воспользуемся обходом в ширину. На очередном шаге будем доставать из очереди вершину v_i и класть в нее вершины v_j, связанные с v_i дорогой, для которых можно улучшить уже посчитанные значения, то есть:

а) либо можно улучшить длину кратчайшего пути.

б) либо длина пути через вершину v_i и уже посчитанное $wayLen[j]$ совпадают, но проехав через вершину v_i можно собрать по дороге больше бензина, чем $maxPetrol[j]$.

Таким образом значения, значения можно улучшить, если:

1) На маршрут через v_i хватает бензина. То есть максимальное количество бензина, который возможно накопить на пути от $start$ до v_i – $maxPetrol[i]$ не меньше длины пути через v_i, то есть не меньше, чем $wayLen[i] + e_{ij}$, где e_{ij} – длина дороги из v_i в v_j.

2) А также верно хотя бы одно из двух:

а) $wayLen[j] > wayLen[i] + e_{ij}$

б) $wayLen[i] + e_{ij} = wayLen[j]$, и при этом $maxPetrol[j] < maxPetrol[i] + petrol[j]$

Если длину кратчайшего пути обновить можно, то:

- $wayLen[j] = wayLen[i] + e_{ij}$
- $maxPetrol[j] = maxPetrol[i] + petrol[j]$
- $prevVertex[j] = i$

Корректность алгоритма доказывается индукцией по длине пути (в смысле количества ребер!) Итого сложность алгоритма $\underline{\mathrm{O}}(N^2)$.