Университет ИТМО
Кафедра ВТ

Алгоритм Беллмана-Форда

Алгоритмы и Структуры Данных

Выполнил: Федоров Сергей
Группа: P3212

Санкт-Петербург
2020 г.

Алгоритм Беллмана-Форда

Алгоритм Беллмана-Форда - алгоритм поиска кратчайших путей. в ориентированном графе, от заданной вершины до всех остальных.

```
Сложность алгоритма по времени: O(|V| * |E|)
Сложность алгоритма по памяти:  O(|V|)
```

Суть алгоритма:

Построим матрицу A_{ij}, где конкретное значение матрицы будет означать кратчайшее расстояние от вершины **s** до вершины **i** при кол-ве ребер в пути не более **j**.

Первый столбец можем инициализировать следующим образом:
$A_{i0} = 0 : i = s, A_{i0} = +\infty : i \neq s$

Стоит заметить, что все пути из s в i, содержащие ровно j ребер состоит из пути длинной **j - 1** и еще одного ребра. Тогда если мы знаем столбцы матрицы $A_{ij} : j \in [0,1...(x-1)]$ то можно легко определить столбец A_{ix}

Матрицу можно заменить на список, тем самым уменьшив сложность алгоритма по памяти с `O(|V|^2)` до `O(|V|)`, но при этом мы тогда теряем возможность нахождения самих путей, а будем хранить лишь расстояние от **s** до остальных вершин.

Псевдно-код:

$$
\begin{array}{l}
\textbf{for } v \in V \\
\quad \textbf{do } d[v] \leftarrow +\infty \\
d[s] \leftarrow 0 \\
\textbf{for } i \leftarrow 1 \textbf{ to } |V| - 1 \\
\quad \textbf{do for } (u, v) \in E \\
\quad\quad \textbf{if } d[v] > d[u] + w(u, v) \\
\quad\quad\quad \textbf{then } d[v] \leftarrow d[u] + w(u, v) \\
\textbf{return } d
\end{array}
$$

Релаксации по каждой вершине выполняются ровно |V| - 1 раз, так как максимальная длинна кратчайшего пути в ребрах будет равна |V| - 1 ребер, если не так, то значит в пути есть циклы, а такой случай мы не рассматриваем.

Доказательство корректности:

Докажем корректность алгоритма по индукции:

Предположение:

После **x** итерций:

- Если **d[i]** (расстояние до **i**) не бесконечность, то она равно какому либо пути от **s** до **i**
- Если существует путь от **s** до **i** с не более чем **x** ребрами, то **d[i]** имеет значение кратчайшего пути из **s** в **i**, с кол-вом ребер не больше **x**

База:

При **x** = 0, тогда по нашей инициализации $d_i = 0 : i = s, d_i = +\infty : i \neq s$, все условия выполняются.

Док-во:

В начале докажем первое утверждение. Каждый раз когда мы обновляем значение в **d** мы делаем сложение двух путей $d[v] \leftarrow d[u] + w(u, v)$, по индукционному предположению $d[u]$ - длинна пути от **s** до **u**. Следовательно $d[v]$ есть длинна какого-то пути от **s** до **v**.

Для доказательства второго утверждения, рассмотрим некоторый кратчайший путь **P** из s до v с количеством ребер в пути не более **j**. Обозначим предпоследнюю вершину в пути **P** буквой **u**. Пусть имеющийся подпуть из **s** в **u**, не кратчайший путь из **s** в **u**, тогда мы можем найти составить такой путь из **P***, который будет короче **P** - противоречие.
По индукционному предположению $d[u]$ на **j - 1** итерации есть кратчайший путь из **s** в **u**, тогда $d[u] + w(u, v)$ есть **P**. На итерации **j** расстояние от **s** до **v** ($d[v]$) будет сравниваться с $d[u] + w(u, v)$ и будет приравнено $d[u] + w(u, v)$ если вдруг окажется больше. Таким образом на итерации **j** $d[v]$ будет равно длине кратчайшего пути **P**.

Оптимизация:

Если при выполнений новой итерации алгоритма минимальные расстояния не изменились, это означает что дальнейшие итерации не найдут более коротких путей, а значит можно прекратить выполнение алгоритма и вернуть результат.

Работа с отрицательными циклами:

Несмотря на то что задача нахождения кратчайшего пути в графе с отрицательными циклами, не имеет смысла, так как можно найти бесконечно малый путь, однако с помощью модифицированного алгоритма Беллмана-Форда можно установить факт наличия отрицательного цикла в графе. Для нахождения такого цикла в графе, достаточно провести еще одну итерацию релаксации после |V| - 1 уже осуществленных и если расстояние до какой-либо из вершин уменьшилось, то это означает что в графе есть отрицательный цикл.