Компьютерные инструменты в образовании, 2017
№ 4: 5–13
УДК: 519.178
http://ipo.spb.ru/journal

НЕОБХОДИМОЕ И ДОСТАТОЧНОЕ УСЛОВИЕ ПРИМЕНИМОСТИ АЛГОРИТМА ДЕЙКСТРЫ

Лебедев С. С.[1], Новиков Ф. А.[1]
[1]Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

Аннотация

Алгоритм Дейкстры является одним из наиболее популярных и фундаментальных алгоритмов решения проблемы поиска кратчайшего пути в ориентированном графе. Хорошо известно, что алгоритм Дейкстры применим к орграфам с неотрицательно взвешенными дугами. Но, как показывают простые наблюдения, существует множество орграфов и даже классов орграфов с отрицательно взвешенными дугами, к которым алгоритм Дейкстры также применим. Таким образом, условие неотрицательности весов дуг является достаточным, но не является необходимым. Необходимое условие применимости алгоритма Дейкстры не было известно. В этой статье мы представляем и доказываем необходимое и достаточное условие применимости алгоритма Дейкстры. Условие основано на введённом нами понятии рекорда пути.

Ключевые слова: *поиск кратчайшего пути, алгоритм Дейкстры, отрицательные веса, необходимое и достаточное условие.*

Цитирование: Лебедев С. С., Новиков Ф. А. Необходимое и достаточное условие применимости алгоритма Дейкстры // Компьютерные инструменты в образовании. 2017. № 4. С. 5–13.

1. ВВЕДЕНИЕ

Алгоритм Дейкстры [1] — одно из наиболее известных и популярных решений задачи о поиске кратчайших путей во взвешенном ориентированном графе (орграфе). Длиной (или весом) пути считается сумма весов входящих в него дуг. Нетрудно показать, что кратчайшие пути из заданного узла s образуют корневое дерево с корнем в узле s [2]. Алгоритм Дейкстры обладает свойством локальной оптимальности: в процессе работы алгоритма дерево кратчайших путей разрастается, но не перестраивается, что позволяет эффективно использовать алгоритм Дейкстры для случая поиска пути из исходного узла s в целевой узел t или в множество целевых узлов.

Задача нахождения кратчайшего пути в орграфе имеет множество практических применений и интерпретаций. В евклидовом графе узлам сопоставляются точки, а дугам — неотрицательные расстояния между ними. Алгоритм Дейкстры находит кратчайшую ломаную, проходящую через заданные точки. Если узлам соответствуют состояния некоторой системы, а вес дуги (u,v) задан как $w(u,v) = -\log P(u,v)$, где $P(u,v)$ — вероятность успешного перехода из состояния v в состояние u, то решение задачи о кратчайшем пути даст наиболее надёжный способ перехода системы из исходного состояния в целевое. Важно заметить, что вес дуги в практической задаче может быть и отрицательным. Например, если при поиске маршрута движения электроавтомобиля весом дуги считать расход энергии аккумулятора, то, двигаясь с горки вниз, можно включить генератор и подзарядить аккумулятор, при этом расход энергии оказывается отрицательным. Таким образом, алгоритм Дейкстры имеет очень широкую область применения, и важно определить условия, при которых использование алгоритма корректно.

Заметим, что алгоритм Дейкстры не всегда находит решение. Во-первых, задача может не иметь решения, если некоторые узлы недостижимы из исходного узла. Во-вторых, решение не определено, если в орграфе есть контуры (замкнутые пути) отрицательного веса, поскольку на таком контуре возможно бесконечно «накручивать» пути все меньшего и меньшего веса. В-третьих, алгоритм Дейкстры может выдать неверное решение, если в графе присутствуют дуги отрицательного веса, в чем можно убедиться на несложных примерах (см. ниже рис. 2). Нами было обнаружено, что в существующей литературе рассмотрение третьего случая далеко не полное.

Сам Дейкстра сформулировал свой алгоритм, употребляя термин «длина ветви», то есть подразумевая, что все дуги имеют положительную нагрузку [1]. Можно одним из хорошо известных способов (см., например, [5]) показать, что применение алгоритма в неотрицательно взвешенном случае действительно корректно. Но также можно привести простые примеры орграфов с отрицательными весами дуг, на которых алгоритм выдаёт верный результат (см. ниже рис. 1).

Разумеется, алгоритм описан и в других источниках, помимо статьи автора. В популярном учебнике [4] проблема применимости не упоминается, а в примерах рассматриваются только неотрицательно взвешенные орграфы. В книге [3] автор делает неоправданно сильное утверждение: «этот алгоритм применяется только к неориентированным и ориентированным графам с неотрицательными весами». В знаменитом учебнике [5] строго доказывается корректность алгоритма Дейкстры для неотрицательно взвешенных ориентированных графов. Полезно также упомянуть книгу [6], которая содержит примечательное изложение вопроса, но условие применимости указано такое же, как в предыдущих источниках.

В этой статье приведены формулировка и доказательство найденного нами необходимого и достаточного условия применимости алгоритма Дейкстры, что позволяет осмысленно использовать его для решения более широкого класса задач, а также устранить методические неточности.

2. АЛГОРИТМ ДЕЙКСТРЫ — ФОРМУЛИРОВКА И ОБСУЖДЕНИЕ

Мы используем простейшее описание алгоритма Дейкстры, следуя, в основном, книге [5]. На вход алгоритма поступает взвешенный ориентированный граф с p узлами, заданный матрицей весов дуг $W : \textbf{array}[1..p, 1..p]\textbf{of real}$, где число $W[u, v]$ означает вес дуги (u, v) из узла u в узел v. Если дуги из узла u в узел v нет, то элемент $W[u, v]$ содержит специальное значение $+\infty$. Начальный узел обозначим s. Предполагается, что в орграфе нет контуров отрицательного веса. На выходе алгоритма вычисляется вектор весов путей $T : \textbf{array}[1..p]\textbf{of real}$, где число $T[v]$ является весом кратчайшего пути из начального узла s в узел v. Если пути из узла s в узел v нет, то $T[v] = +\infty$. Сами кратчайшие пути накапливаются в векторе предшествования $P : \textbf{array}[1..p]\textbf{of}\ 0..p$, где элемент $P[v]$ хранит номер узла, непосредственно предшествующего узлу v на кратчайшем пути из узла s в узел v. Если пути из узла s в узел v нет, то $P[v] = 0$. Кроме того, алгоритм использует вспомогательную структуру данных — контейнер Q, в который можно поместить номер узла v вызовом процедуры $\mathrm{putQ}(v)$ и извлечь номер узла v с минимальным значением $T[v]$ вызовом функции $v := \mathrm{extractMinQ}()$.

Заметим, что выбор использованных структур данных (исходная матрица весов W, векторы результатов T и P, контейнер Q) не является однозначным. Вместо матрицы весов W можно использовать списки смежности или массив дуг, контейнер Q может быть реализован различными способами: в виде массива, списка, двоичной кучи и т. д., сохранять результаты можно не в отдельных векторах, а с помощью дополнительных отметок на узлах и дугах. Все эти «программные хитрости» существенным образом влияют на практическую трудоёмкость алгоритма Дейкстры, но не влияют на его теоретическую применимость. Целью этой статьи является только исследование применимости алгоритма, поэтому мы используем наипростейшие струкутры данных и оставляем вопросы эффективной реализации без обсуждения, тем более, что эти вопросы достаточно обсуждены, например, в учебнике [8].

В указанных обозначениях алгоритм Дейкстры можно записать на псевдокоде следующим образом:

{инициализация}
for $v = 1$ **to** p **do**
  $T[v] := +\infty$ {в начале пути неизвестны}
  $\mathrm{putQ}(v)$ {и все узлы помещаются в контейнер Q}
end for
$T[s] := 0$ {нулевой минимальный путь из s в s известен}
{основной цикл}
{пока контейнер Q не пуст}
while $Q \neq \varnothing$ **do**
  $u := \mathrm{extractMinQ}()$ {извлечение узла с минимальным значением $T[u]$}
  if $T[u] = +\infty$ **then**
    stop {остальные узлы недостижимы из s}
  end if
  {для всех узлов}
  for $v = 1$ **to** p **do**
    {если узел v смежный с узлом u}
    {и узел v находится в контейнере Q }
    {и путь через узел u короче}
    if $W[u, v] \neq +\infty$ **and** $v \in Q$ **and** $T[v] > T[u] + W[u, v]$ **then**
      $T[v] := T[u] + W[u, v]$ {вес пути от s к v через u}
      $P[v] := u$ {узел u предшествует v на кратчайшем пути из s}
    end if
  end for
end while

Действие по пересчёту значений векторов T и P в двух последних строках алгоритма традиционно называется *релаксацией*, или ослаблением дуги (u, v).

Рассмотрим примеры работы алгоритма Дейкстры. На рис. 1 представлен случай, когда алгоритм правильно находит дерево кратчайших путей при наличии дуг отрицательного веса. На рисунке узлы обозначены кружками с буквами, вес дуги указан на дуге, рядом с узлом выписано значение из массива T, узел, выбираемый функцией extractMinQ, выделен жирной окружностью, а узлы, извлеченные из контейнера Q, залиты серым цветом.

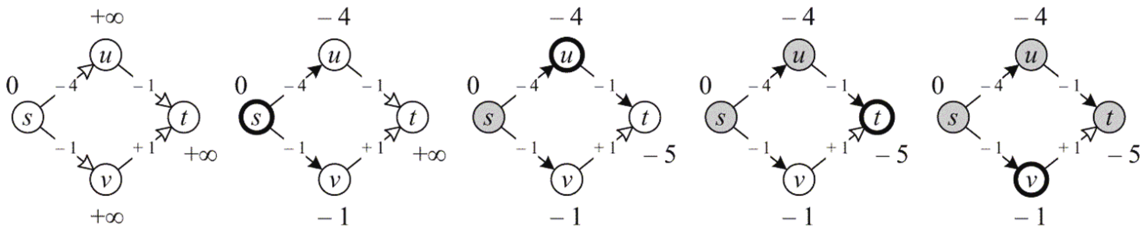

Рис. 1. Правильная работа алгоритма Дейкстры

Однако в случае орграфа, представленного на рис. 2, алгоритм работает неправильно и не находит минимального пути в узел t.

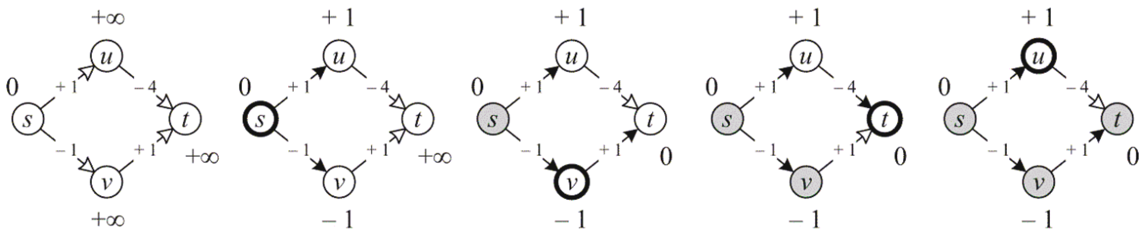

Рис. 2. Неправильная работа алгоритма Дейкстры

В следующем разделе выявлена причина, по которой алгоритм Дейкстры «ошибается» во втором случае.

3. НЕОБХОДИМОЕ И ДОСТАТОЧНОЕ УСЛОВИЕ ПРИМЕНИМОСТИ АЛГОРИТМА ДЕЙКСТРЫ

Заметим, что если в некоторый путь входит контур положительного веса, то такой путь не может быть кратчайшим, поскольку контур можно исключить и длина пути уменьшится. Контуры отрицательного веса запрещены постановкой задачи. Поэтому далее мы рассматриваем только простые пути без самопересечений, то есть без контуров. Назовём *рекордом* пути ($R(M)$, где $M = \langle u, w\rangle$ — путь) максимальный вес начального отрезка этого пути:

$$R(< u, w >) = \max\{W(\langle u, v\rangle) : v \in M \setminus u\},$$

где $W(\langle u, v\rangle)$ — вес пути. Отдельно отметим, что $v \neq u$, то есть рекорд не может достигаться в начальном узле по определению, откуда следует, что

$$\forall \langle u, w\rangle : R(< u, w\rangle) \geq W(\langle u, w\rangle) \geq \Omega(u, w),$$

где $\Omega(u, w)$ — вес кратчайшего пути из узла u в узел w.

Рассмотрим сеть путей $P(s, t)$ из узла s в узел t. Заметим, что для любого промежуточного узла v в сети $P(s, t)$ определены сети $P(s, v)$ и $P(v, t)$, причём они являются сегментами исходной сети:

$$\forall v \in P(s, t) : P(s, v) \subset P(s, t) \wedge P(v, t) \subset P(s, t).$$

Далее можно взять промежуточный узел в одном из сегментов и получить ещё два сегмента, причем один из них уже не будет инцидентен исходным узлам s и t. Таким образом, можно декомпозировать сеть на сегменты вплоть до отдельных дуг. Пути в сегментах могут не пересекаться, а могут пересекаться и даже слипаться. Если начальный узел некоторого сегмента имеет только одну исходящую дугу, то такой сегмент называется *безальтернативным*. Все пути безальтернативного сегмента слиплись в первой дуге. Если из начального узла исходит более одной дуги, то сегмент называется *альтернативным* или *развилкой*. Очевидно, что при поиске кратчайших путей в сегменте безальтернативные участки путей можно пропускать и начинать поиск с первой развилки.

Теорема. *Алгоритм Дейкстры на взвешенном орграфе без контуров отрицательного веса строит дерево кратчайших путей с корнем в узле s тогда и только тогда, когда для каждого узла $t \in G \setminus s$, достижимого из узла s, в сети $P(s,t)$ для каждой развилки $P(w,t)$ рекорд кратчайшего пути $\langle w,t\rangle$ строго меньше, чем рекорд любого не кратчайшего пути $\langle w,t\rangle$.*

Доказательство. Заметим, что, когда узел u извлечён из контейнера, все узлы на пути $\langle s,u\rangle$ уже были извлечены ранее, и поэтому оценка извлечённого узла не превосходит рекорда найденного пути к нему: $T[u] \le R(\langle s,u\rangle)$.

Достаточность. Покажем по индукции, что на каждой итерации при извлечении узла u путь $M_0 = \langle s,u\rangle$, определяемый значением $P[u]$ (см. псевдокод алгоритма), кратчайший, то есть $W(\langle s,u\rangle) = \Omega(s,u)$. Тем самым покажем, что на каждой итерации извлечённые узлы и пути к ним образуют частичное дерево кратчайших путей из узла s, и по завершении работы алгоритма построено дерево кратчайших путей из узла s ко всем достижимым узлам.

База: на первой итерации всегда выбирается узел s, который образует корень дерева. Длина кратчайшего пути $\Omega(s,s) = 0$, поскольку контуры запрещены.

Индукционное предположение: пусть на предыдущих итерациях построено частичное дерево кратчайших путей и пусть из контейнера Q выбран узел u с оценкой $T[u]$, причём $T[u] < +\infty$ и узел u получил свою оценку в результате релаксации дуги (v,u). Если путь $\langle s,\ldots,v,u\rangle$ не единственный, то в сети $P(s,u)$ рассмотрим первую (считая от узла s) развилку $P(w,u)$ и возможные альтернативные пути: $M_0 = \langle w,\ldots,v,u\rangle$; M_1, в котором все узлы, кроме последнего, уже исключены из Q на предыдущих итерациях; M_2, в котором есть узлы, принадлежащие Q (рис. 3).

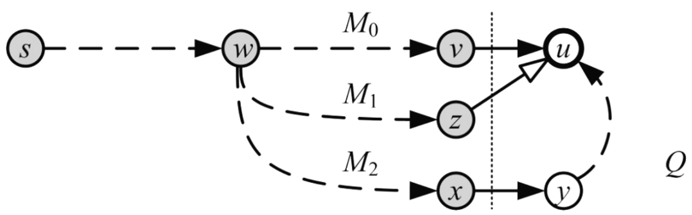

Рис. 3. Рассматриваемые варианты альтернативных путей

В случае пути M_1 обе дуги (v,u) и (z,u) уже участвовали в релаксации, и в построенном дереве сохранена та, которая даёт меньшую оценку, поэтому путь M_1 не оказывает влияния на построенное дерево, и его можно не учитывать.

Далее доказательство от противного. Пусть $W(s,\ldots,w,M_2) < W(s,\ldots,w,M_0)$. По условию теоремы, это означает, что $R(M_2) < R(M_0)$. Пусть $u_0 \in M_0$ — место достижения рекорда пути M_0. Тогда $T[u_0] = T[w] + R(M_0)$ (узел u_0 может быть узлом u, v или любым промежуточным узлом). Рассмотрим ту итерацию, когда был выбран узел u_0, и пусть при этом u_2 — первый ещё не извлечённый узел на пути M_2 (узел u_2 может быть узлом y, x или некоторым промежуточным узлом). Тогда

$$T[u_0] = T[w] + R(M_0) > T[w] + R(M_2) \geq T[u_2],$$

что противоречит тому, что узел u_0 извлечён раньше узла u_2. Доказательство очевидным образом распространяется на все последующие развилки.

Необходимость. Нужно показать, что если алгоритм построил дерево кратчайших путей, то в любой развилке рекорд кратчайшего пути меньше рекорда не кратчайшего пути, или, в контрапозитивной форме, если существует развилка, в которой рекорд кратчайшего пути больше либо равен рекорду не кратчайшего пути, то алгоритм «ошибётся» и построенное дерево не будет деревом кратчайших путей.

Рассмотрим сначала случай, когда рекорд кратчайшего пути строго больше рекорда не кратчайшего пути. Пусть алгоритм Дейкстры построил дерево, и пусть существует узел u, такой что в него ведут два пути: путь M_0' — кратчайший и M_2' — не кратчайший: $W(M_0') < W(M_2')$. Пусть $P(w,u)$ — первая развилка этих путей, M_0 и M_2 — продолжения путей после развилки соответственно (рис. 4), причём $R(M_0) > R(M_2)$.

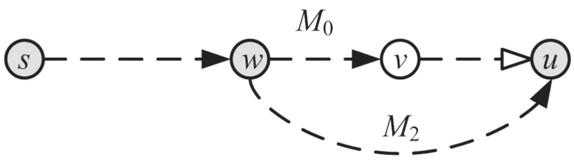

Рис. 4. Кратчайший путь M_0 и не кратчайший путь M_2

Рассмотрим узел $v \in M_0$, в котором достигается рекорд, то есть $T[v] = T[w] + R(M_0)$. Тогда

$$\forall x \in M_2 : T[x] \leq T[w] + R(w,x) < T[w] + R(M_0) = T[v],$$

и алгоритм выберет все узлы пути M_2, включая узел u, раньше, чем узел v, а значит какие-то дуги на пути $\langle v,u\rangle$ не будут участвовать в релаксации и кратчайший путь не будет найден.

Рассмотрим теперь случай, когда рекорд кратчайшего пути равен рекорду не кратчайшего пути. В такой ситуации корректность работы алгоритма Дейкстры зависит от воли случая, то есть от того, какой из узлов с равными значениями оценки T будет выбран. Например, в орграфе на рис. 5 с начальным узлом s на первом шаге оба узла u и v получают одинаковые оценки $T[u] = T[v] = 1$. Если на втором шаге будет выбран узел v, то произойдет релаксация дуги (v,u), и будет найдет кратчайший путь $\langle s,v,u\rangle$. Если же на втором шаге будет выбран узел u, то релаксации дуги (v,u) не произойдет, и кратчайший путь не будет найден.

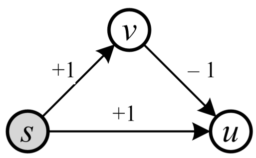

Рис. 5. Возможный ложный выбор алгоритма Дейкстры

Мы считаем, что в тех случаях, когда мы не можем быть уверены в корректной работе алгоритма, в рамках теоремы следует считать работу алгоритма некорректной.

Теорема доказана. □

4. СЛЕДСТВИЯ

Доказанная теорема позволяет легко обосновать несколько полезных частных случаев.

Следствие 1. *Алгоритм Дейкстры корректен на классе положительно взвешенных орграфов.*

Доказательство. Положительно взвешенный орграф очевидно не содержит контуров отрицательного веса. Далее заметим, что из положительности весов следует, что рекорд любого пути достигается в его конечном узле, следовательно рекорд кратчайшего пути меньше рекордов альтернативных или равен им в конечном узле. □

Аналогично можно показать справедливость и других общеизвестных свойств алгоритма Дейкстры.

Также нетрудно распространить алгоритм на некоторые встречающиеся на практике специальные графы. Назовём *столбчатым* положительно взвешенный орграф, расширенный дополнительными тупиковыми дугами отрицательного веса из его узлов. На практике такой граф можно сопоставить задаче о поиске кратчайшего пути с бонусом за остановку.

Следствие 2. *Алгоритм Дейкстры корректен на классе столбчатых орграфов.*

Доказательство. Аналогично первому следствию, рекорд достигается либо в конечном узле, либо за один узел до конечного, и тогда путь из него до конечного не содержит развилок. □

Назовём *гирляндой* орграф, состоящий из подграфов A, B и так далее, причём из A в B ведёт ровно одна дуга (a, b), из B в следующую часть графа также ведёт ровно одна дуга и так далее. Соединительные дуги гирлянды могут быть как положительными, так и отрицательными. Графы такого рода часто возникают при программировании компьютерных игр.

Следствие 3. *Алгоритм Дейкстры корректен на классе гирляндных графов, у которых все подграфы удовлетворяют условию теоремы.*

Доказательство. Пусть для определённости начальный узел s принадлежит подграфу A. Тогда в подграфе A дерево кратчайших путей строится корректно по условию, в подграфе B дерево кратчайших путей имеет общий начальный путь $\langle s, \ldots, a, b \rangle$, и далее оно корректно достраивается из узла b, и так далее. □

5. ЗАКЛЮЧЕНИЕ

Одной из исходных целей данного исследования было устранение методической неполноты в курсе дискретной математики, поставленном на кафедре «Прикладная математика» СПбГПУ [2]. Эта цель была достигнута полностью.

Изначально у авторов была обоснованная гипотеза, что алгоритм Дейкстры может иметь значительно большую область применимости, чем принято считать. Гипотеза подтвердилась частично, поскольку полученный нами класс графов, с одной стороны, действительно шире, чем неотрицательно взвешенные орграфы, но, с другой стороны, уже, чем требуется для большинства задач на графах. Но то обстоятельство, что доказанное нами условие не только достаточное, но и необходимое, позволяет поставить точку в этом вопросе.

Вопрос практической применимости условия остаётся открытым. С одной стороны, для графов общего вида известны алгоритмы, не сильно уступающие алгоритму Дейкстры по времени работы, особенно в простых случаях (например алгоритм Левита [7]). С другой стороны, задача, где применение именно классического алгоритма Дейкстры даст наилучший результат, вполне может проявиться со временем.

Список литературы

1. *Dijkstra E. W.* A note on two problems in connexion with graphs // Numerische Mathematik. 1959. Vol. 1. P. 269–271.
2. *Новиков Ф. А.* Дискретная математика. 3-е изд. СПб.: Питер, 2017.
3. *Anderson J. A.* Discrete Mathematics with Combinatorics (2nd Edition). Prentice Hall, 2003.
4. *Levitin A.* Introduction to the design & analysis of algorithms (3rd Edition). Addison-Wesley, 2012.
5. *Cormen T. H., Leiserson Ch. E., Rivest R. L., Stein C.* Introduction to Algorithms (3rd Edition). The MIT Press, 2009.
6. *Липский В.* Комбинаторика для программистов. М.: Мир, 1988.
7. *Романовский И. В.* Дискретный анализ. 4-е издание. СПб.: Невский диалект, 2008.
8. *Алексеев В. Е., Таланов В. А.* Графы. Модели вычислений. Струкутры данных. Нижний Новгород: Изд-во ННГУ, 2005.

Поступила в редакцию 10.06.2017, окончательный вариант — 21.07.2017.

Computer tools in education, 2017
№ 4: 5–13
http://ipo.spb.ru/journal

THE NECESSARY AND SUFFICIENT CONDITION FOR DIJKSTRA'S ALGORITHM APPLICABILITY

Lebedev S. S.[1], Novikov F. A.[1]

[1]SPbPU, Saint-Petersburg, Russia

Abstract

Dijkstra's algorithm is one of the most popular and fundamental algorithms solving the shortest path problem in directed graphs (digraphs). It is well known that Dijkstra's algorithm is applicable to digraphs with non-negative weighted arcs. But as simple observations show there are many digraphs and even classes of digraphs with negatively weighted arcs for which the Dijkstra's algorithm is also applicable. Thus the non-negative weight of the arcs condition is not a necessary condition but only a sufficient one. The necessary condition for applicability of Dijkstra's algorithm was unknown. In this paper, we present and prove a necessary and sufficient condition for the applicability of Dijkstra's algorithm. The condition is based on the notion of a path's record that we introduce.

Keywords: *shortest path problem, Dijkstra's algorithm, negative weight, necessary and sufficient condition.*

Citation: S. S. Lebedev & F. A. Novikov, "Neobkhodimoe i dostatochnoe uslovie primenimosti algoritma Deikstry" [The Necessary and Sufficient Condition for Dijkstra's Algorithm Applicability], *Computer tools in education*, no. 4, pp. 5–13, 2017 (in Russian).

Received 10.06.2017, the final version — 21.07.2017.

Lebedev Sergei Sergeevich, student, SPbPU, ssergejslebedev@gmail.com
Novikov Fedor Alexandrovich, technical science doctor, professor of the applied math departement of SPbPU, fedornovikov51@gmail.com

---

Лебедев Сергей Сергеевич,
студент СПбПУ,
ssergejslebedev@gmail.com

Новиков Федор Александрович,
доктор технических наук, профессор
кафедры прикладной математики СПбПУ,
fedornovikov51@gmail.com