Алгоритм нахождения кратчайшего пути

Нахождение ***кратчайшего пути*** на сегодняшний день является жизненно необходимой задачей и используется практически везде, начиная от нахождения оптимального маршрута между двумя объектами на местности (например, *кратчайший путь* от дома до школы), в системах автопилота, для нахождения оптимального маршрута при перевозках, и т.п.

Большое значение данная тема имеет для программистов.

Кратчайший путь рассматривается при помощи графа.

Поиск кратчайшего пути ведется между двумя заданными вершинами в графе.

Результатом является *путь*, то есть последовательность вершин и ребер, и его *длина*.

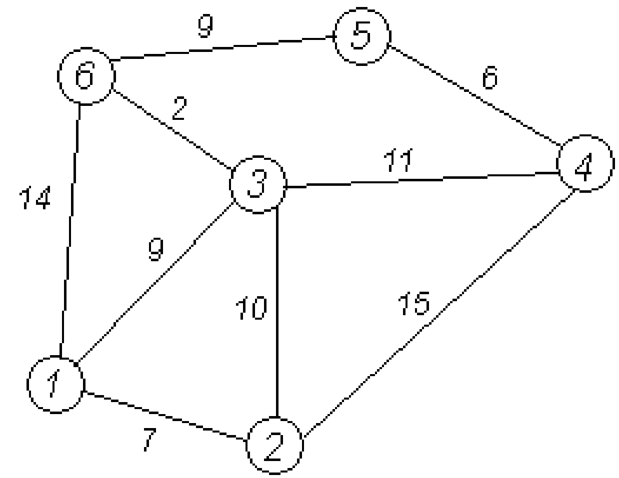
Для решения задачи нахождения кратчайшего пути можно использовать *алгоритм Дейкстры* — алгоритм на графах, изобретённый нидерландским ученым Э. Дейкстрой в 1959 году.

Алгоритм Дейкстры

Рассмотрим выполнение алгоритма на примере графа, показанного на рисунке.

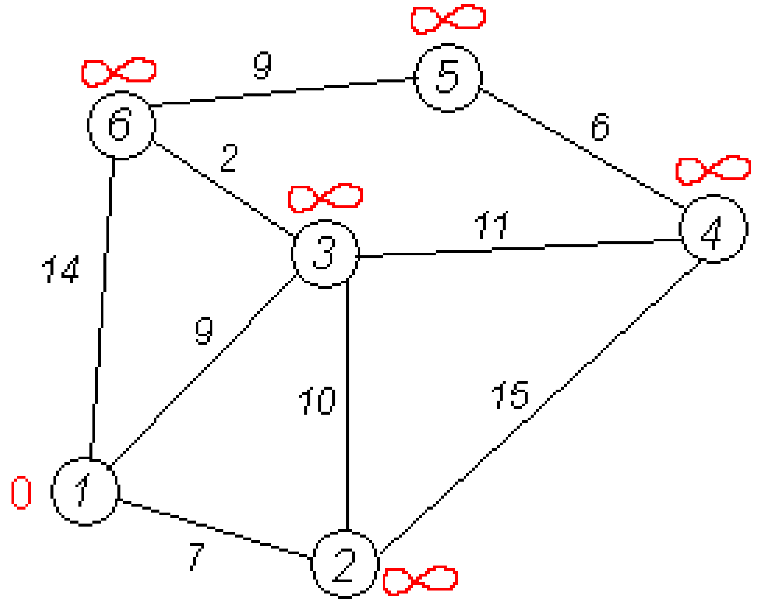
Пусть требуется найти кратчайшие расстояния от 1-й вершины до всех остальных.

Кружками обозначены вершины, линиями – пути между ними (ребра графа). В кружках обозначены номера вершин, над ребрами обозначен их вес – длина пути. Рядом с каждой вершиной красным обозначена метка – длина кратчайшего пути в эту вершину из вершины 1.

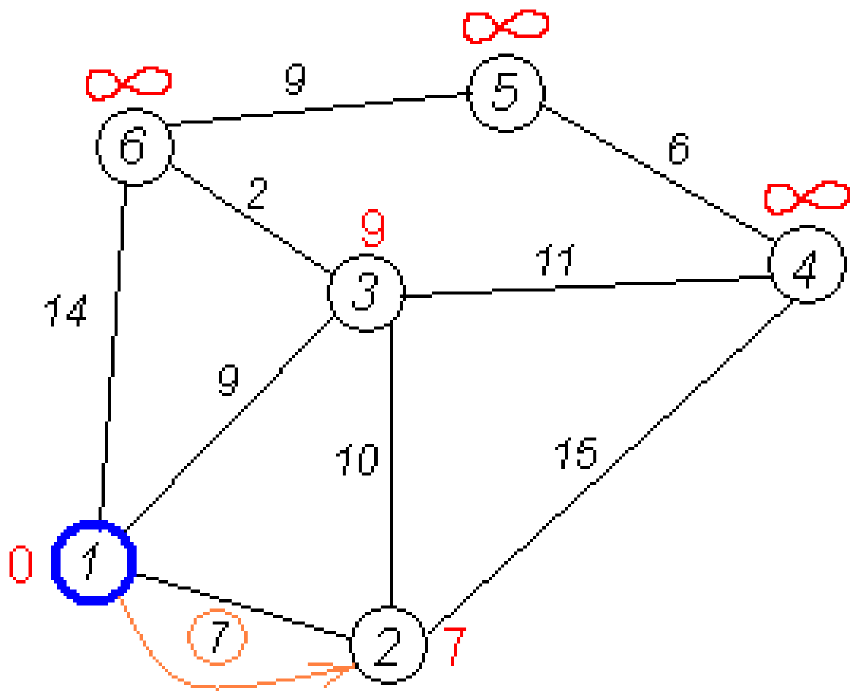


Первый шаг. Рассмотрим шаг алгоритма Дейкстры для нашего примера. Минимальную метку имеет вершина 1. Её соседями являются вершины 2, 3 и 6.

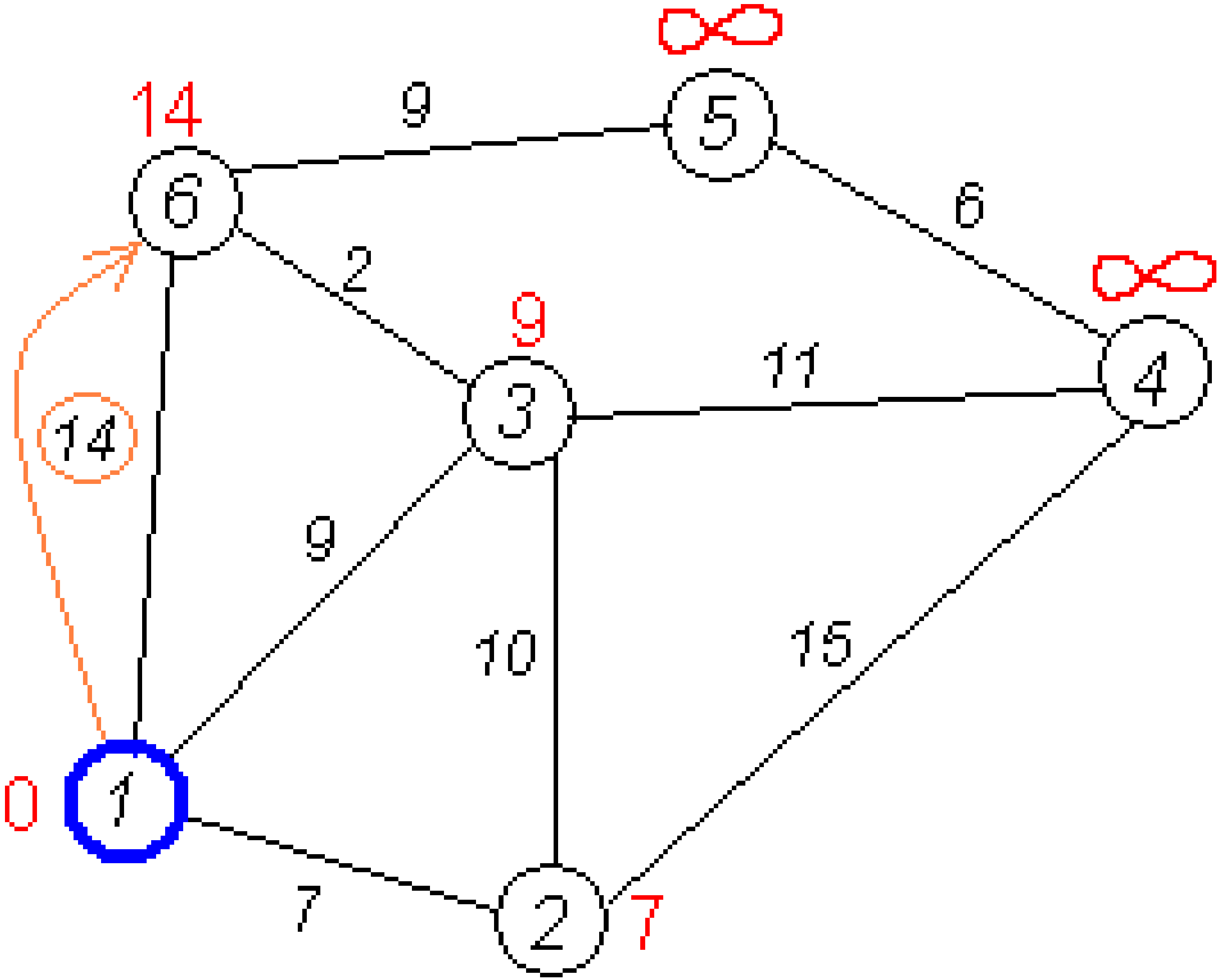
Метка самой вершины 1 полагается равной 0, метки остальных вершин обозначим . Это отражает то, что расстояния от вершины 1 до других вершин пока неизвестны.



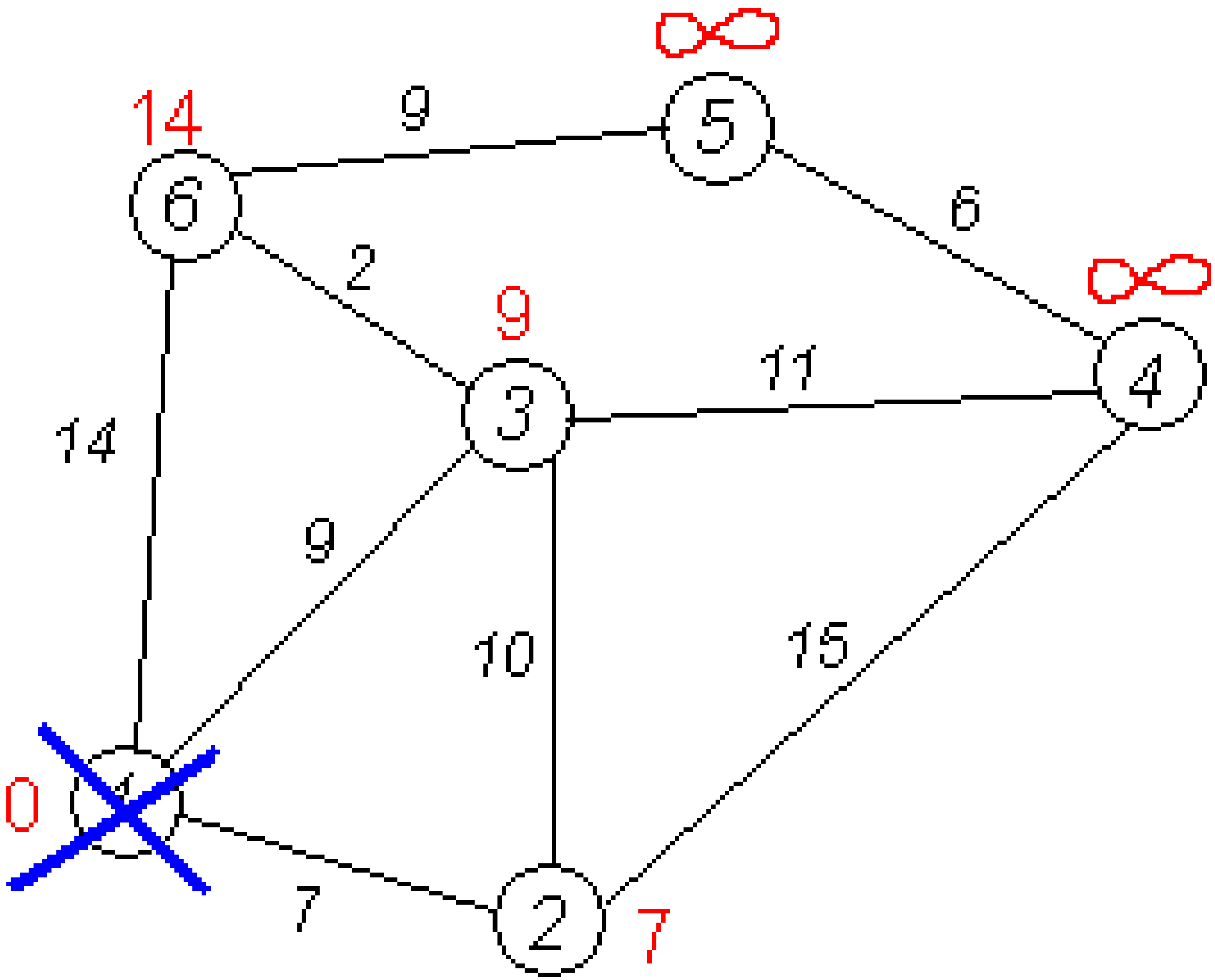
Первый по очереди сосед вершины 1 — вершина 2, потому что длина пути до неё минимальна.
Длина пути в неё через вершину 1 равна сумме метки вершины 1 и длины ребра, идущего из 1-й в 2-ю, то есть 0 + 7 = 7.
Новая метка 2-й вершины равна 7.



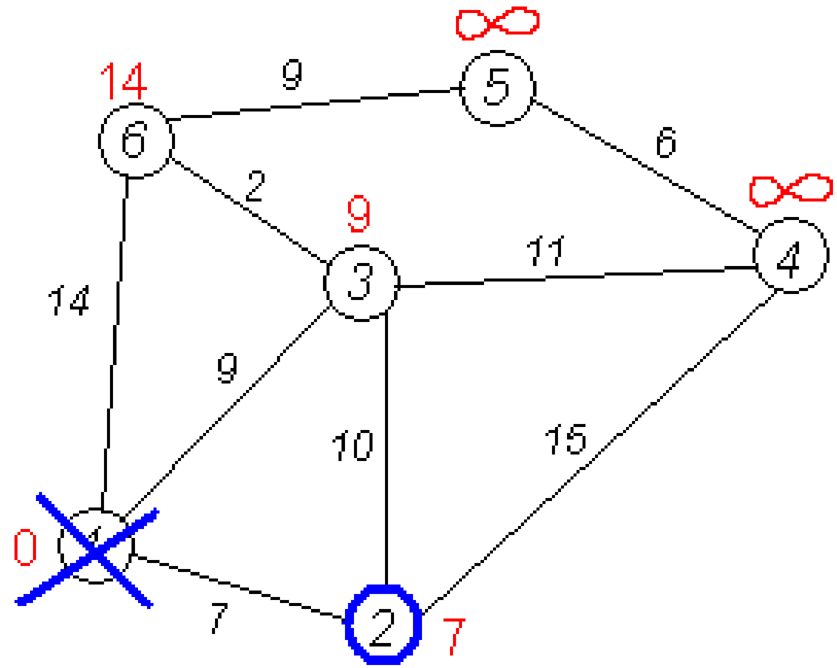
Аналогично определим метки вершин 3 и 6.



Все соседи вершины 1 проверены. Текущее минимальное расстояние до вершины 1 считаем равным **0. Вершину 1 вычеркиваем.**



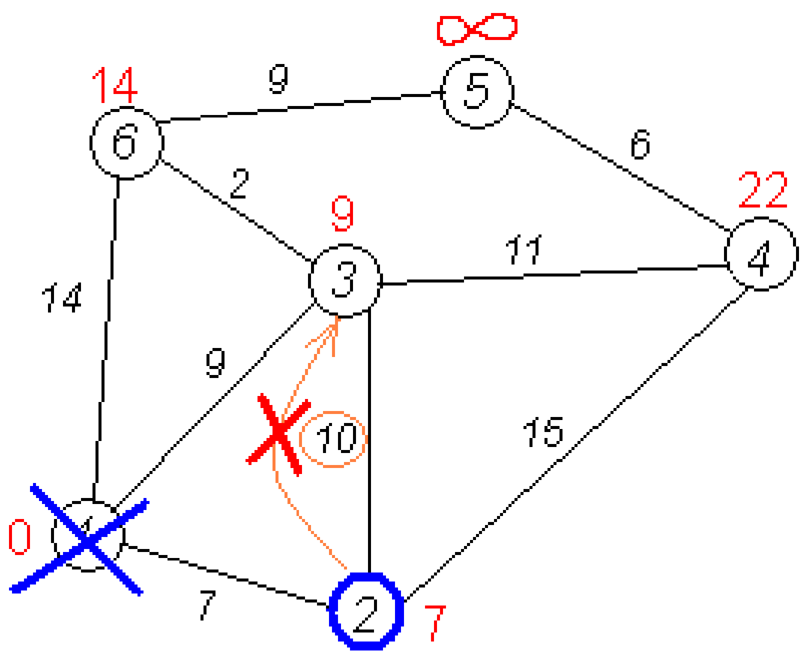
Второй шаг. Снова находим «ближайшую» из непосещённых вершин. Это вершина 2 с меткой 7.



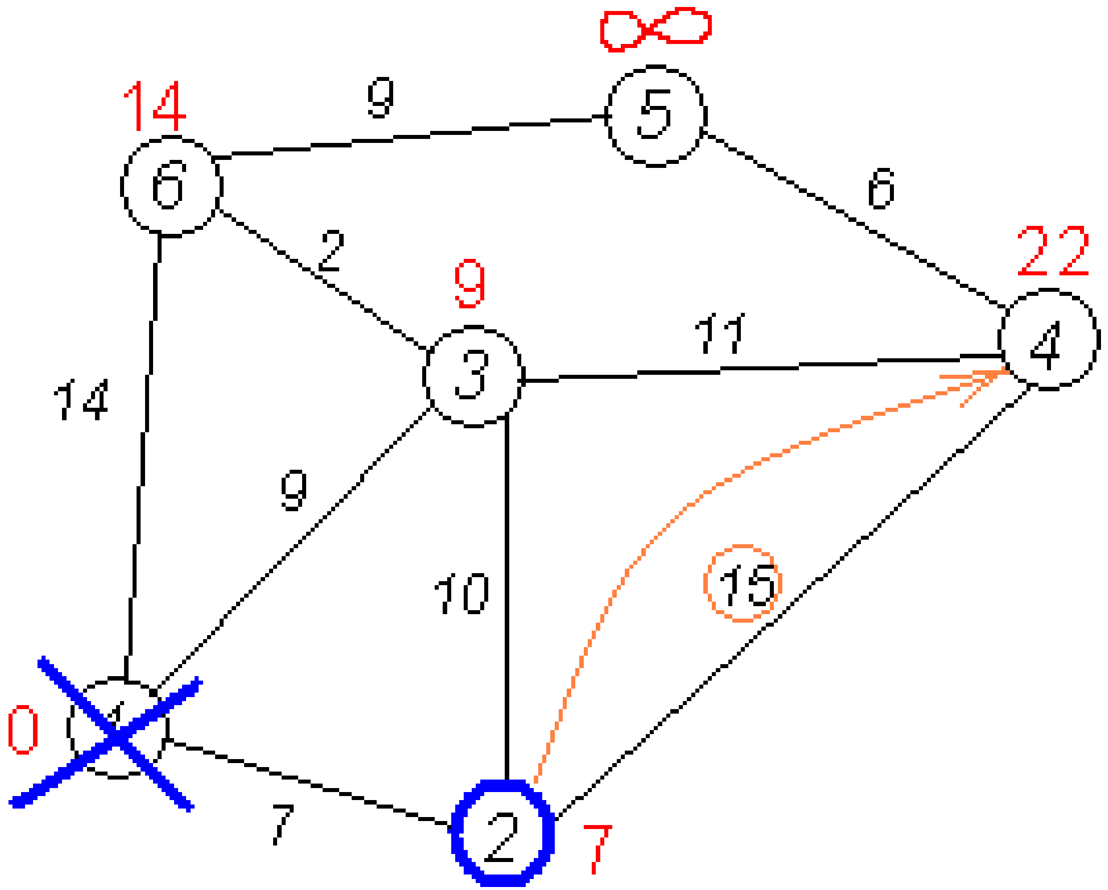
Соседями вершины 2 являются вершины 1, 3 и 4.

Первый (по порядку) сосед вершины 2 — вершина 1. Но она уже посещена, поэтому с 1-й вершиной ничего не делаем.

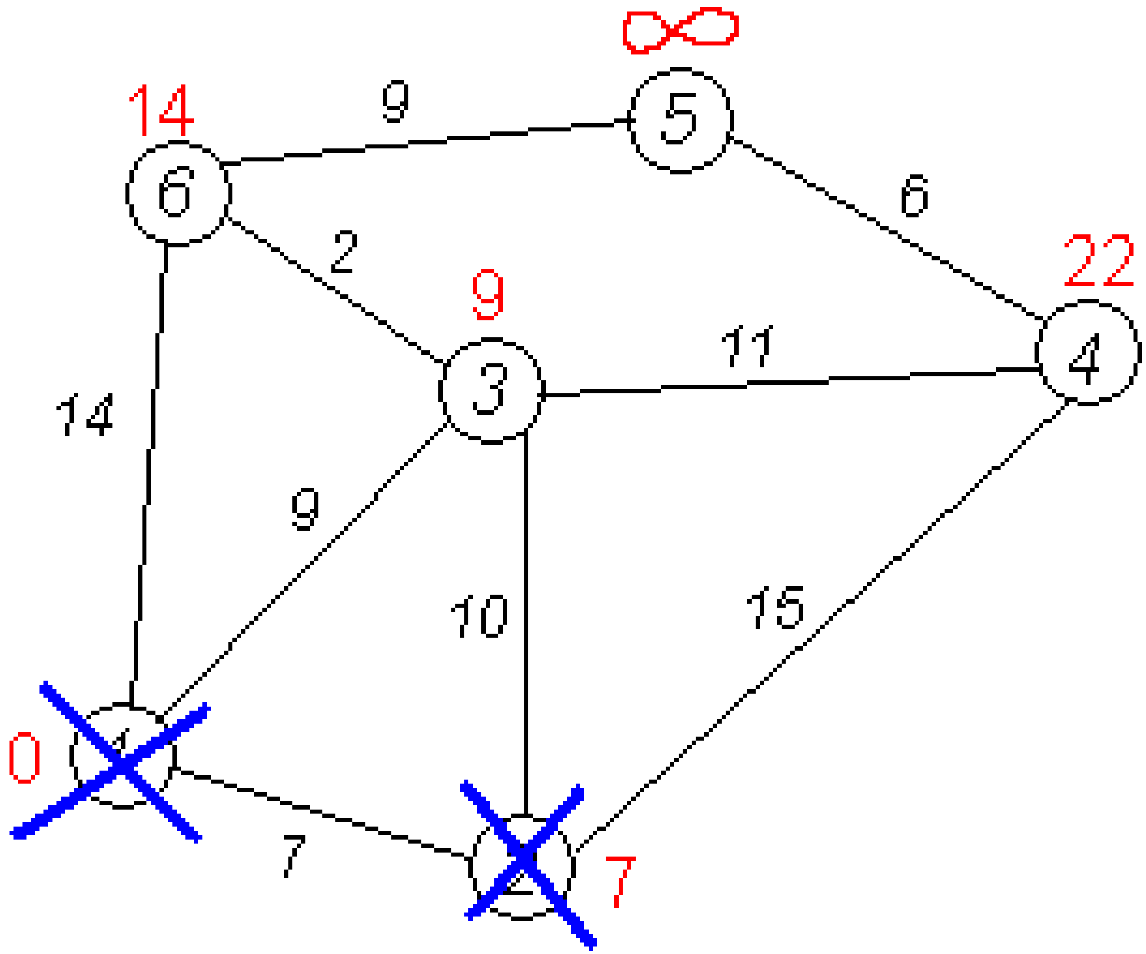
Следующий сосед вершины 2 — вершина 3, так как имеет минимальную метку из вершин, отмеченных как не посещённые.
Если идти в неё через 2, то длина такого пути будет равна 7 + 10 = 17.
Но текущая метка третьей вершины равна 9, а это меньше 17, поэтому метка не меняется.



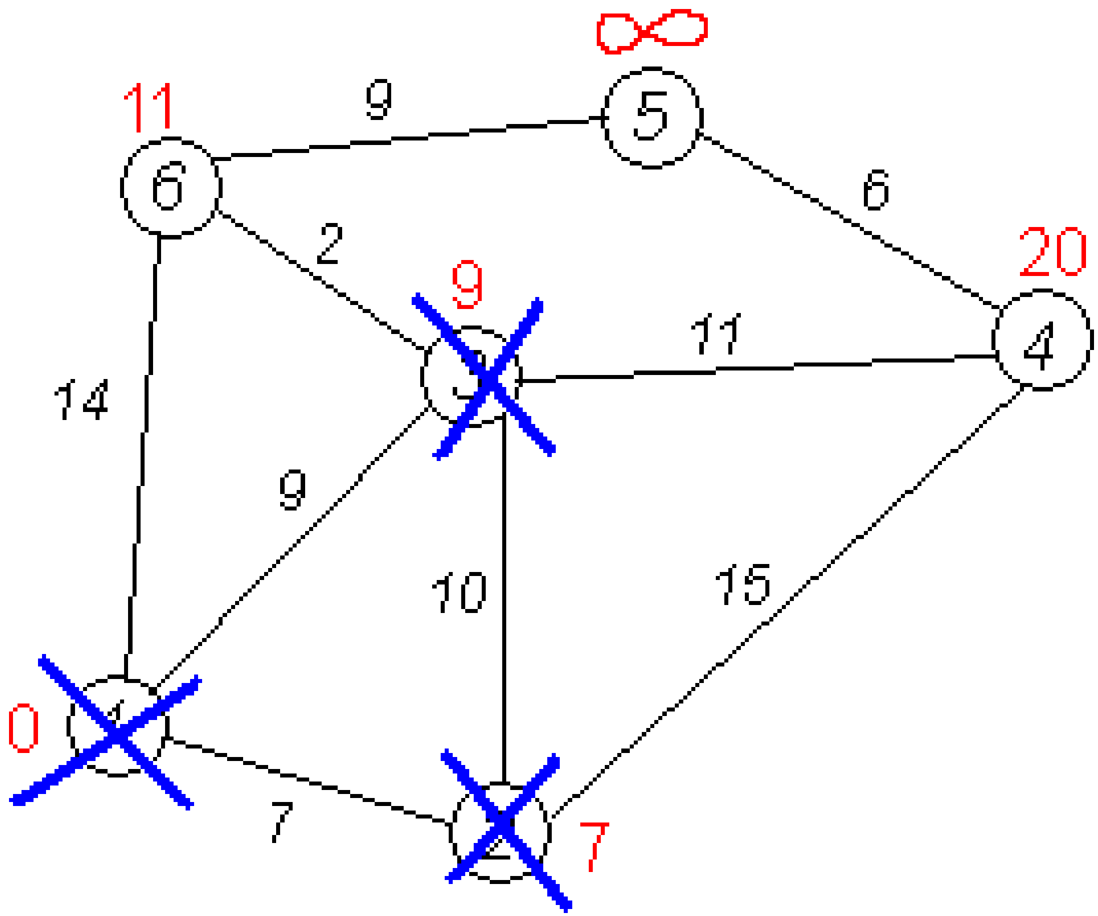
Ещё один сосед вершины 2 — вершина 4. Если идти в неё через 2-ю, то длина такого пути будет 7 + 15 = 22. Устанавливаем метку вершины 4 равной 22.



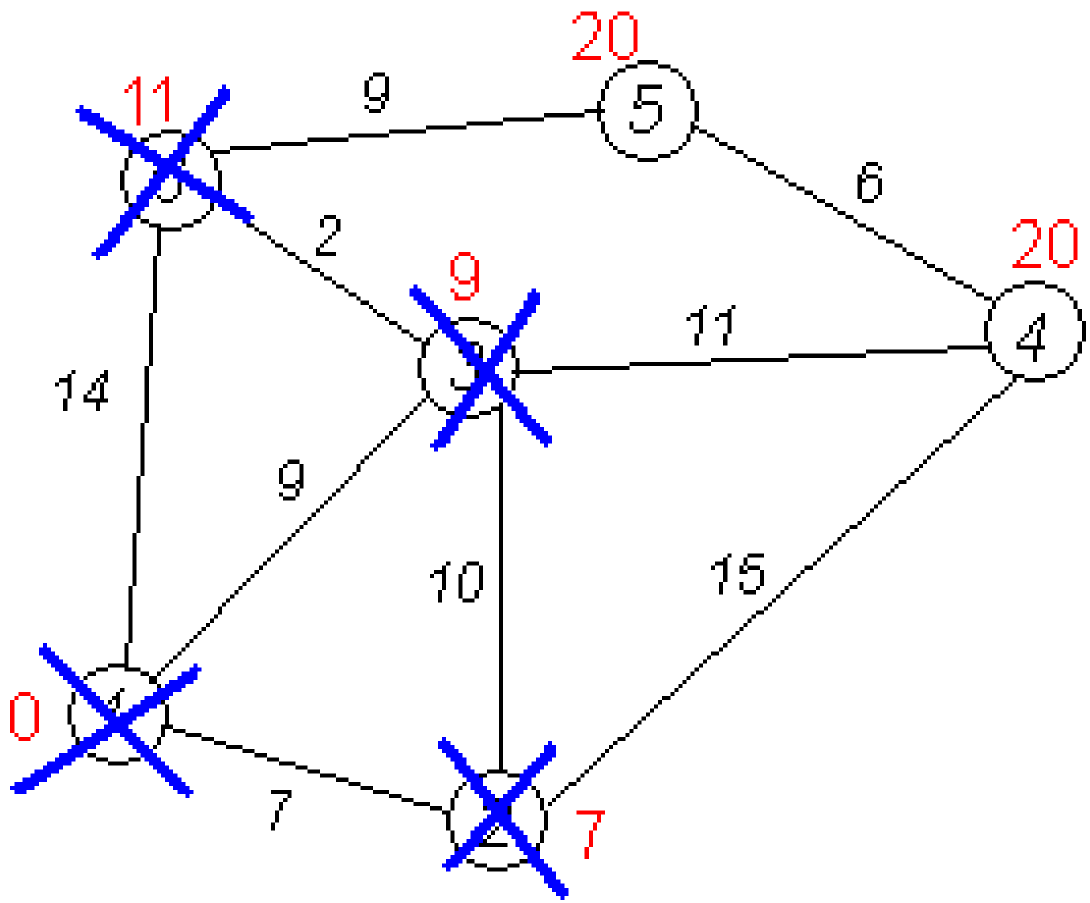
Все соседи вершины 2 просмотрены, помечаем её как посещённую.

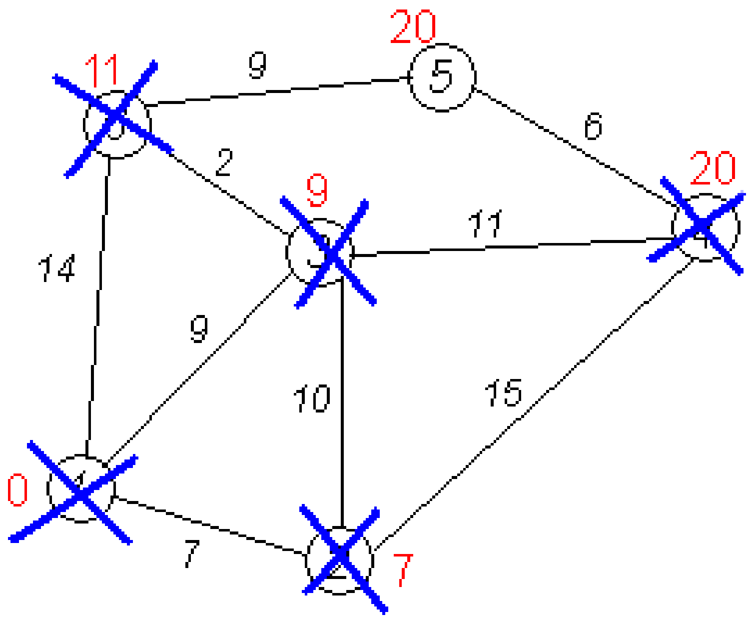


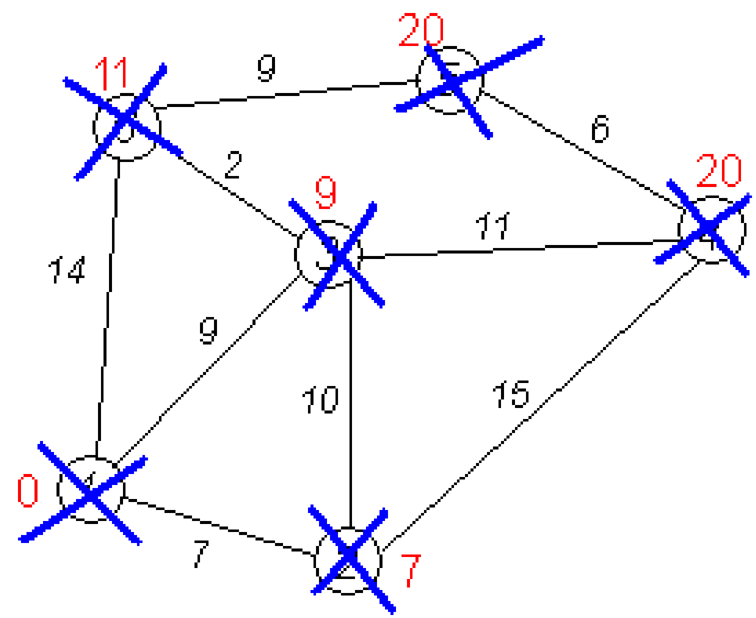
Третий шаг. Повторяем шаг алгоритма, выбрав вершину 3.
Метка вершины 6 равна 9+2=11, Метка вершины 4 равна 9+11=20, так как 20 меньше, 7+15=22



Повторяем шаг алгоритма для оставшихся вершин. Следующая вершина 6. Из вершины 6 можем попасть в вершину 5. Определим метку вершины 5. 9+9+2=20.



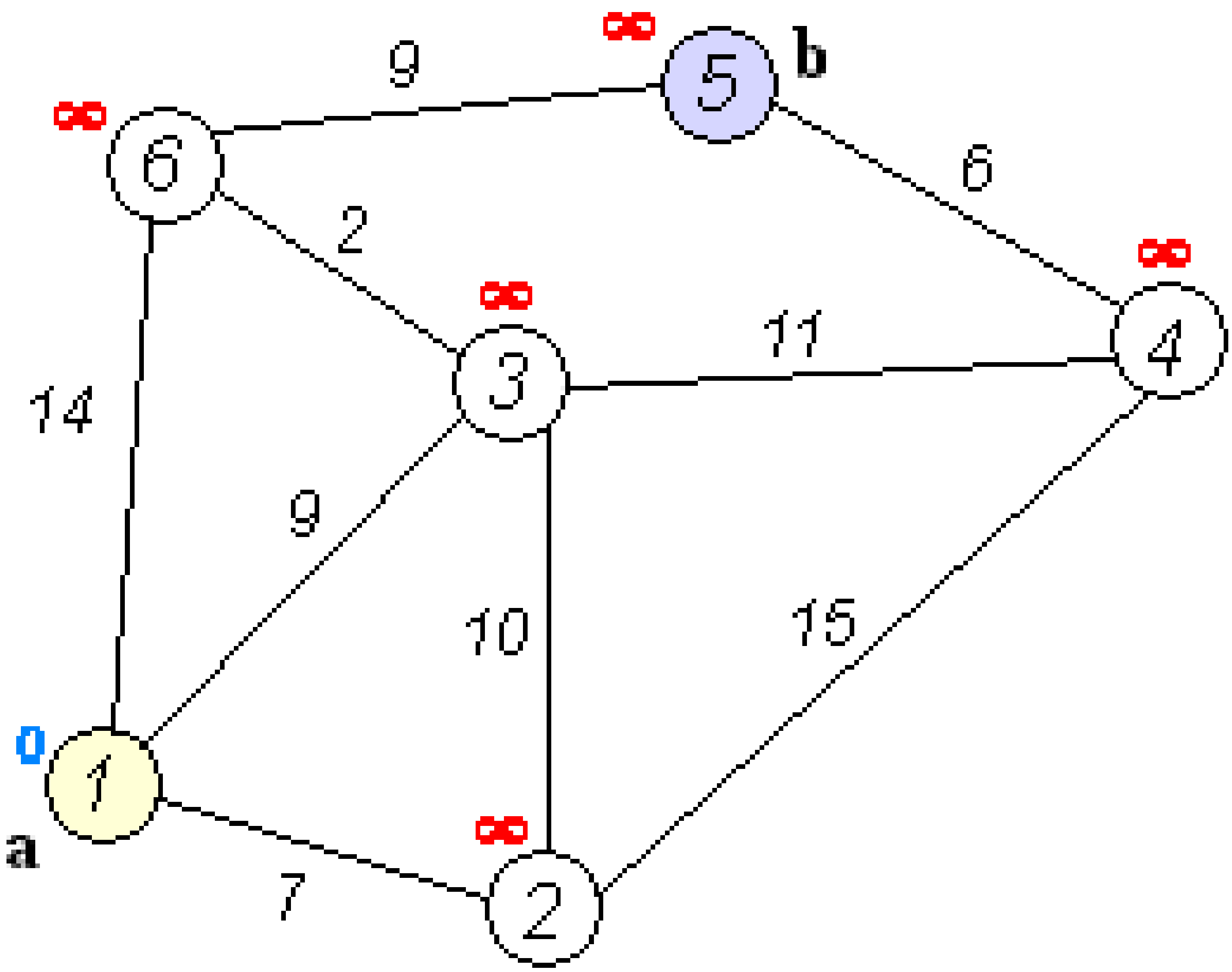




Алгоритм заканчивает работу, когда нельзя больше обработать ни одной вершины.

Результат работы алгоритма виден на последнем рисунке: **кратчайший путь от вершины 1 до 2-й составляет 7, до 3-й — 9, до 4-й — 20, до 5-й — 20, до 6-й — 11.**

Повторим алгоритм еще раз!



Выполни задания

Найдите кратчайшие расстояния от 1 -й вершины до всех остальных для графа, представленного на рисунке.

Надо найти кратчайшие расстояния от склада до каждой строительной площадки. Какова длина кратчайшего пути от склада до строительной площадки 1? (**На графе склад обозначен вершиной 7**).