Компоненты сильной связности (КСС) орграфа *G* – это его максимальные сильно связные подграфы.

Если вершина не связана с другими, то считается, что она сама образует КСС.

Орграф, который получается стягиванием в одну вершину каждой КСС, называется фактор-графом, или конденсацией орграфа *G*.

Для выделения КСС орграфа можно в качестве основы алгоритма использовать метод поиска в глубину.

В качестве примера рассмотрим Алгоритм Косараджу–Шарира (Kosaraju, Sharir)

1 этап: запускается поиск в глубину на обращении графа

2 этап: запускается поиск в глубину на исходном графе в порядке, определяемом списком, полученным на первом этапе (в обратном порядке)

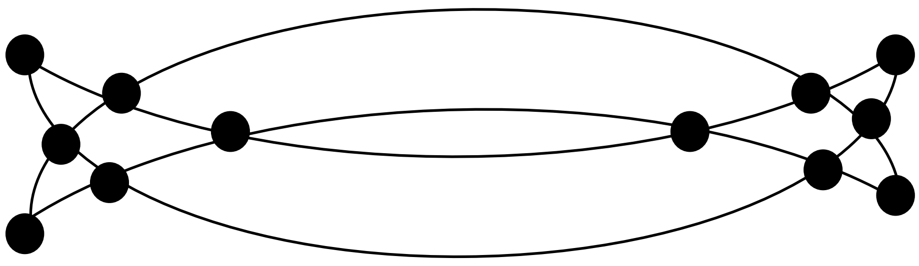
3.13 Эйлеровы графы

Цикл в графе называется эйлеровым, если он содержит все ребра графа.

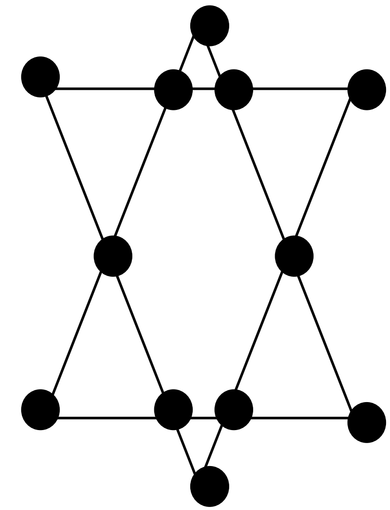
Связный граф, в котором существует эйлеров цикл, называется эйлеровым графом.

Эйлеровой цепью (или путем) является цепь (путь), которая включает все ребра (дуги) графа по одному разу.

Собственная эйлерова цепь – это эйлерова цепь, которая не является эйлеровым циклом.



сабли Магомета

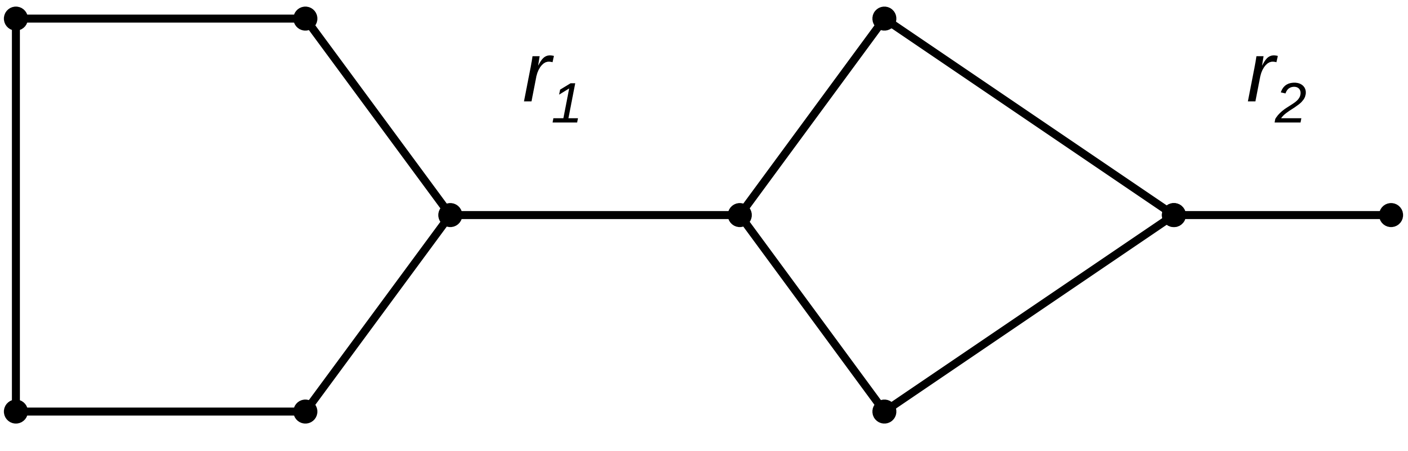


звезда Давида

Теорема Эйлера-2. ***Связный граф является эйлеровым тогда и только тогда, когда степени всех его вершин четны.***

Граф имеет собственную эйлерову цепь (путь) $\Leftrightarrow$ когда он связный и ровно две его вершины имеют нечетную степень.

Мостом (или перешейком) называется такое ребро графа *G*, удаление которого увеличивает число связных компонент. Если ребро *r* принадлежит некоторому циклу *C*, то оно не может быть мостом.



алгоритм Флери:

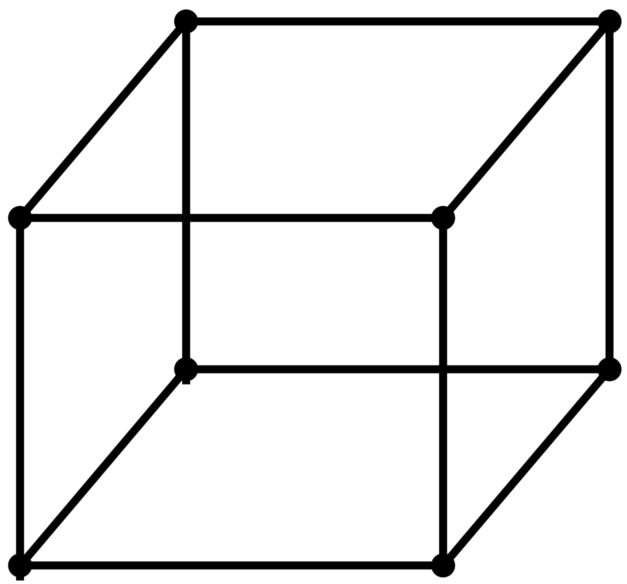
1) Движение начинается из произвольной вершины графа; идем по ребрам, включая эти ребра в эйлерову цепь и удаляя их из графа.
2) В очередной вершине выбираем путь по перешейку только в том случае, если нет пути по циклу.
3) Если в графе остаются ребра, которые нельзя использовать для продолжения имеющегося пути, то следует начать строить простой замкнутый цикл из уже пройденной вершины и инцидентного ей ребра, если последнее ранее не использовалось.

3.14 Гамильтоновы графы

Гамильтонова цепь (путь) – это простая цепь (путь), которая проходит через каждую вершину (узел) графа ровно по одному разу.

Гамильтонов цикл – это простой цикл, который проходит через каждую вершину графа.

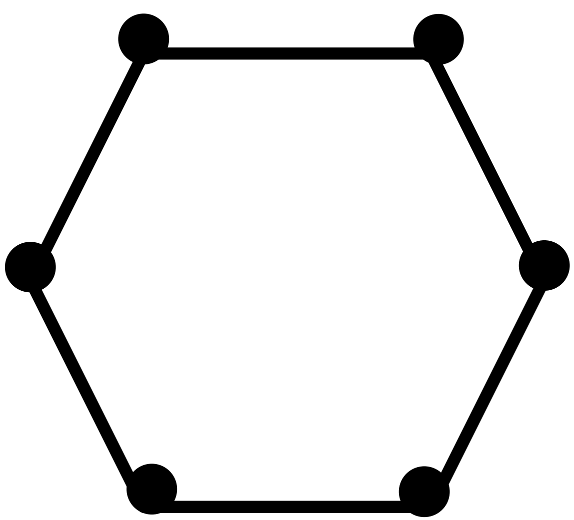
Граф, содержащий гамильтонов цикл, называется гамильтоновым графом.



Не существует четкого критерия для определения, является ли граф гамильтоновым.

Теорема (Оре): Пусть G –связный граф с n вершинами (n>2). Если для любых несмежных вершин u и v deg(u)+deg(v) $\geq$ n, то граф Гамильтонов

Теорема (Дирака): Пусть G –связный граф с n вершинами (n>2). Если для любой вершины v deg(v) $\geq$ n/2, то граф Гамильтонов



n=6, $\forall$v deg(v)=2

Оре: 2+2=4 $\geq$6 – не выполняется

Дирака:

2 $\geq$ 6/2 – не выполняется

3.15 Минимальный остов

Вес остовного дерева взвешенного графа равен сумме весов, приписанных его ребрам.

Минимальным остовным деревом называется остовное дерево графа с минимальным весом.

Математическая формулировка задачи: во взвешенном связном графе G(V,E) найти минимальное остовное дерево T(V,E’).

Алгоритм Краскала (Cruskal)

Первоначально множество E’ пустое, V – множество вершин графа. Два следующих действия выполняются до тех пор, пока это возможно.

1) Выбрать ребро минимального веса в исходном графе G, не принадлежащее множеству E’, так, что его добавление в E’ не создает цикла в дереве T.
2) Добавить это ребро во множество ребер E’.

3.16 Кратчайшие пути

Рассматривается взвешенный граф (орграф) $G(V,E)$

Вес обозначает длину (или стоимость) пути из одного конца ребра в другой. Если из вершины v_i нет ребра (дуги) в вершину v_j, то вес ребра (v_i,v_j) считается равным ∞. Для ребер, являющихся петлями (диагональ матрицы смежности), их веса будем считать равными 0.

Алгоритм Дейкстры

Находит кратчайшее расстояние от одной фиксированной вершины до другой и указывает сам путь, длина которого равна этому расстоянию.

Для примера будем искать кратчайшее расстояние от вершины v_1 до вершины v_n.

Каждой вершине v_k ставится в соответствие упорядоченная пара (m,v_r),первоначально все вершины помечены $(\infty,0)$ и имеют статус временных.

Алгоритм состоит из следующих шагов:

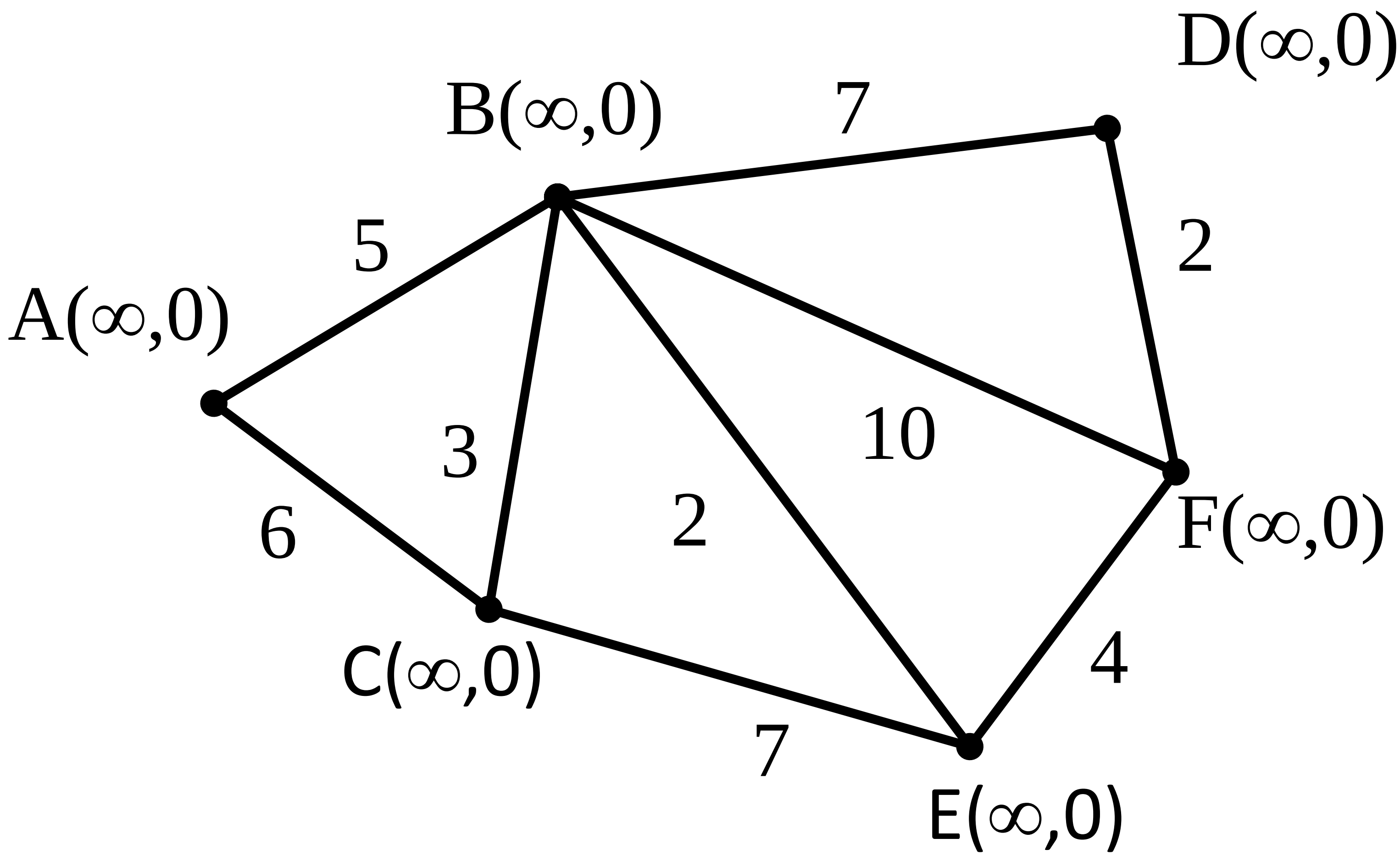
1. Начать в вершине $v_1(\infty,0)$, заменить ее метку на $v_1(0,0)$ и сделать эту вершину постоянной.
2. Пока v_n не станет постоянной вершиной, выполнять следующие шаги:

а)Если вершина $v_k(m,v_r)$ стала постоянной, то для всех смежных с ней временных вершин v_j вычислить $m+d(v_k,v_j)$ и сравнить с текущим расстоянием, которым помечена вершина v_j. Если полученная сумма меньше, то текущее расстояние заменить ею, а вторую координату заменить на v_k.

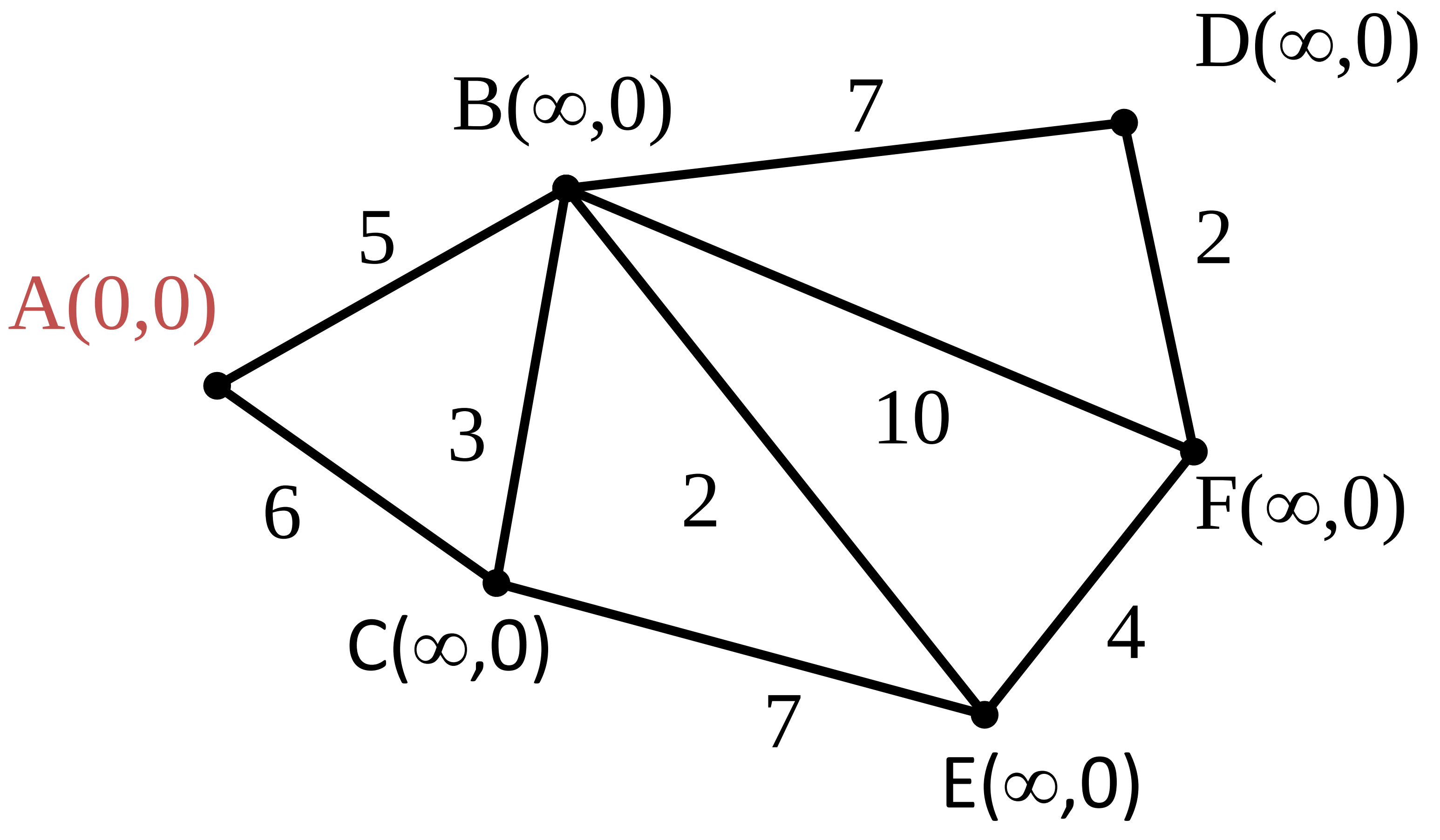
b) Найти минимум из расстояний, приписанных временным вершинам. Первую из таких вершин сделать постоянной.

3. Когда v_n станет постоянной вершиной, то расстояние, приписанное ей – это и есть длина кратчайшего пути. Сам путь можно отследить, если пройти в обратную сторону –от вершины v_n к v_1 по вторым координатам меток вершин.

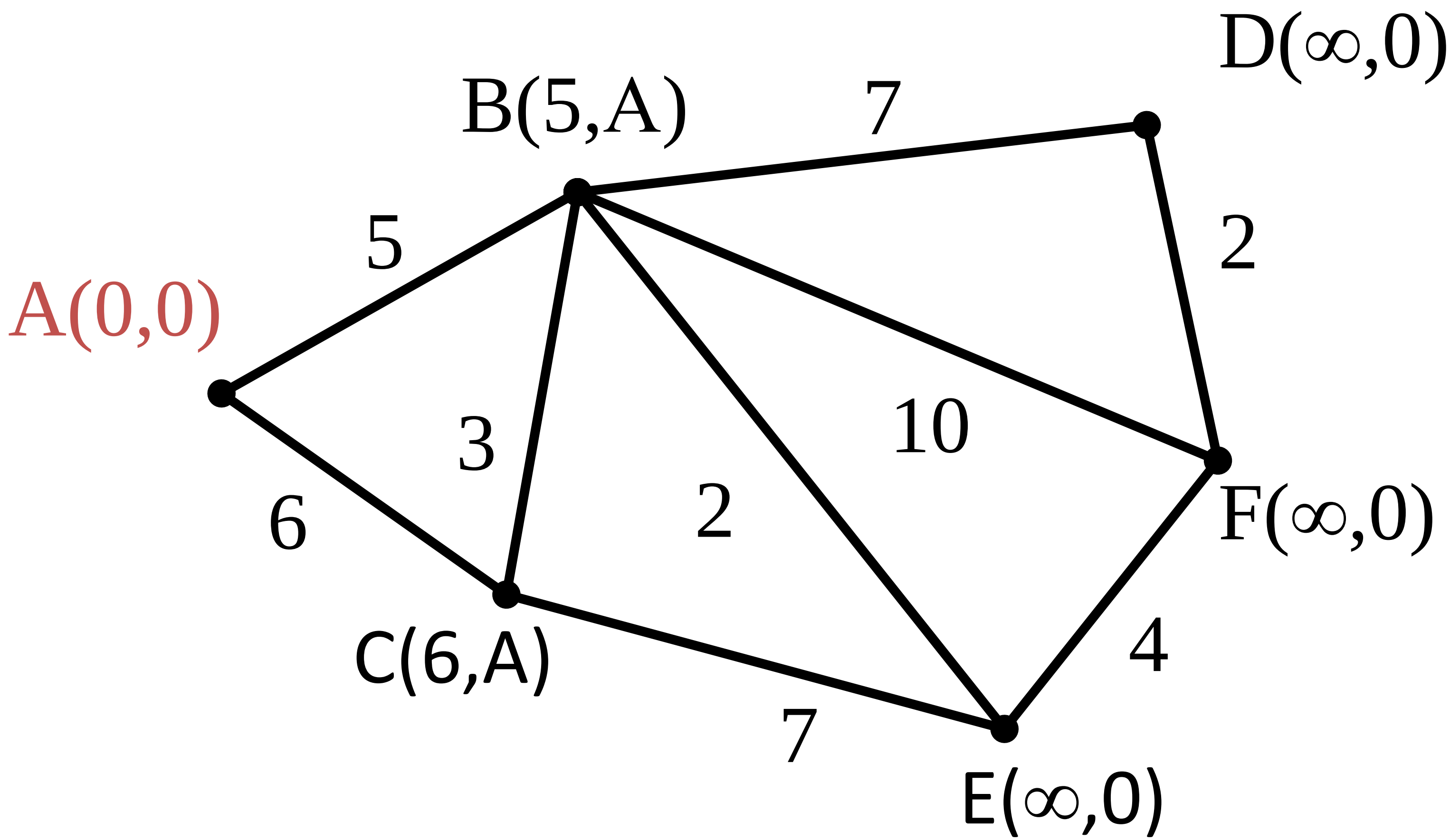
Пример. Найти кратчайший путь из вершины A в вершину F



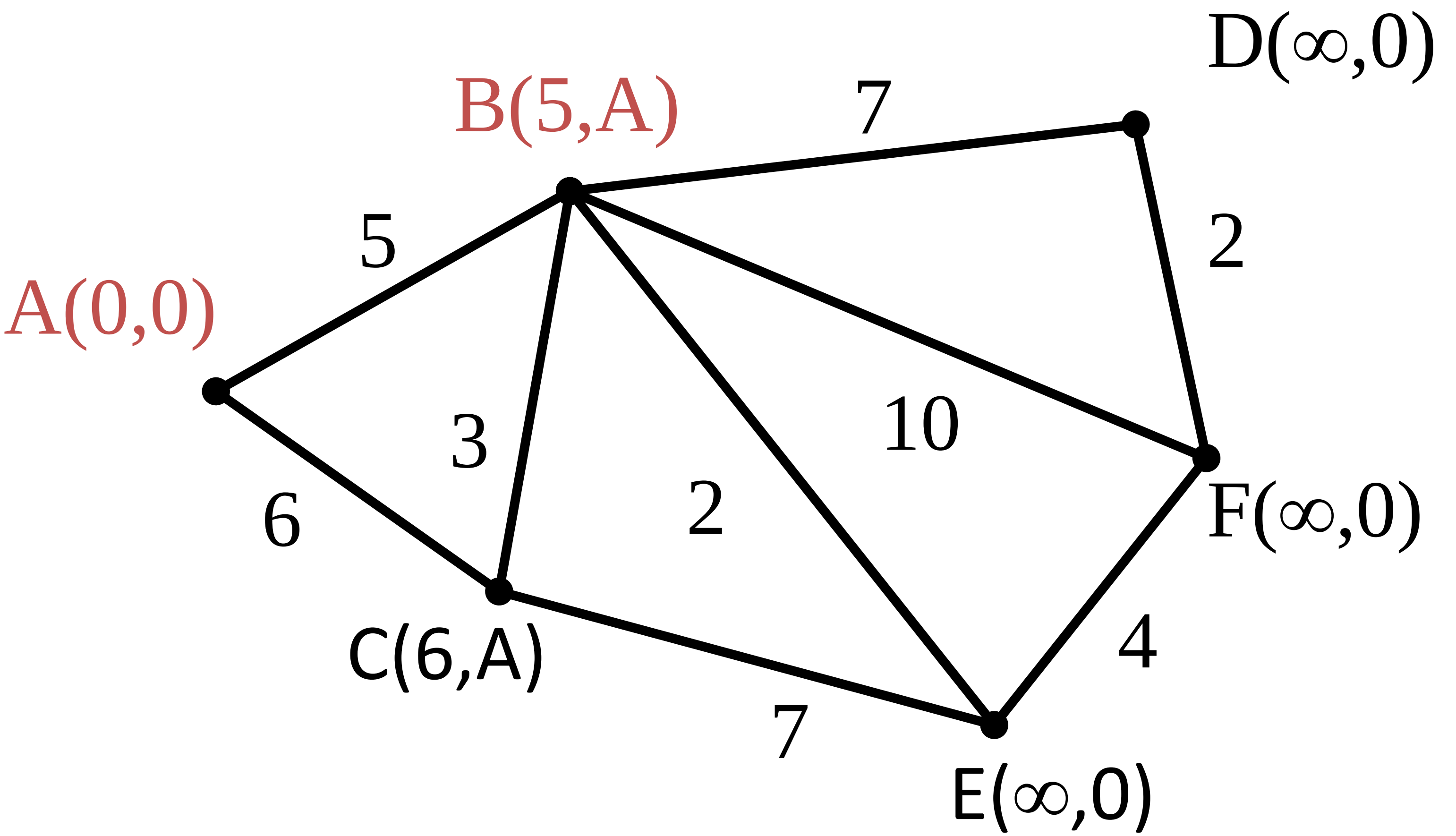
Вершина A стала постоянной



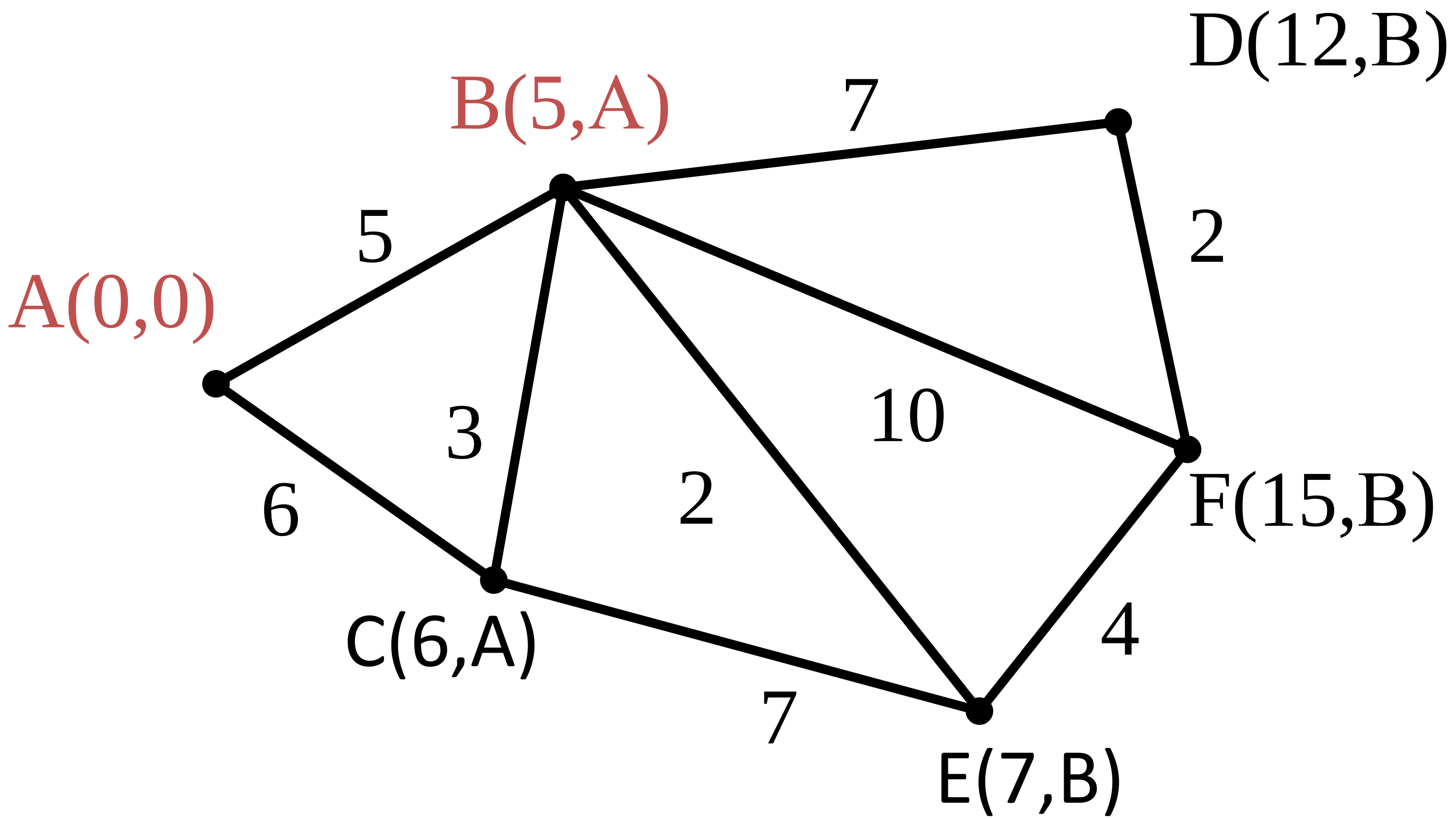
Пересчитали расстояние для смежных с А вершин: В и С.



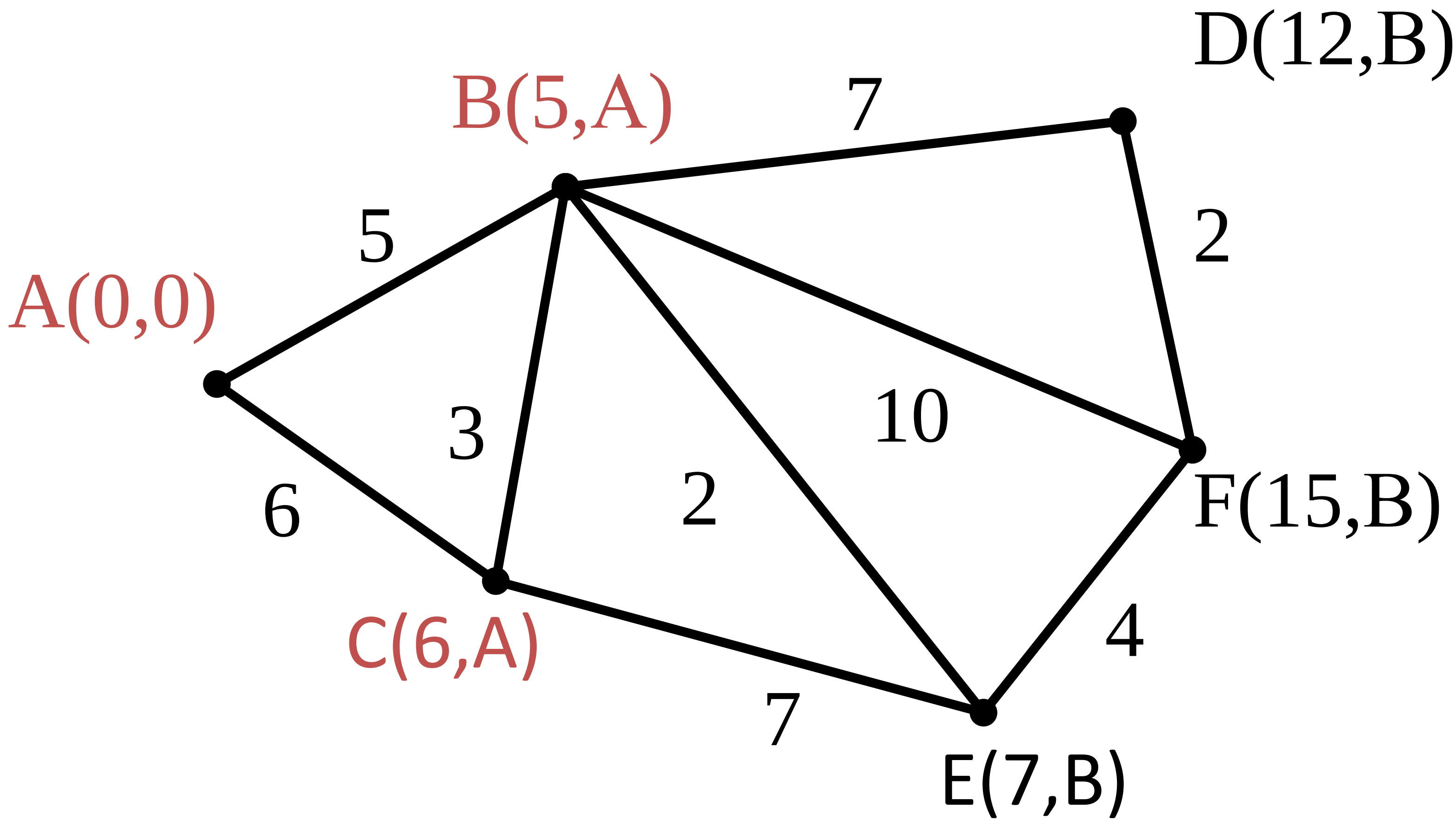
Минимальное из расстояний 5 $\Rightarrow$ вершина B(5,A) становится постоянной



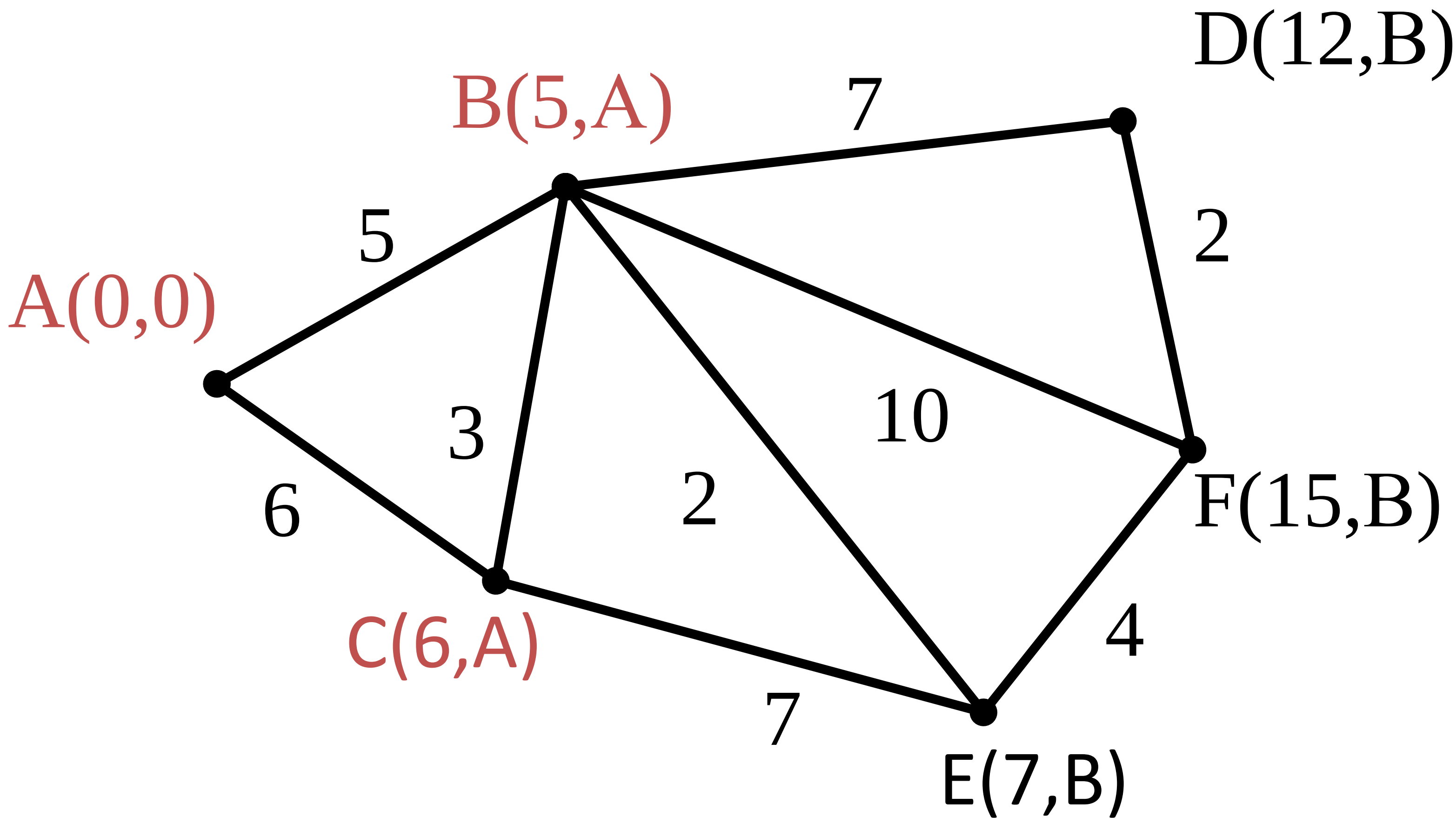
Вычисляем расстояния от B до смежных с ней C, E, F, D



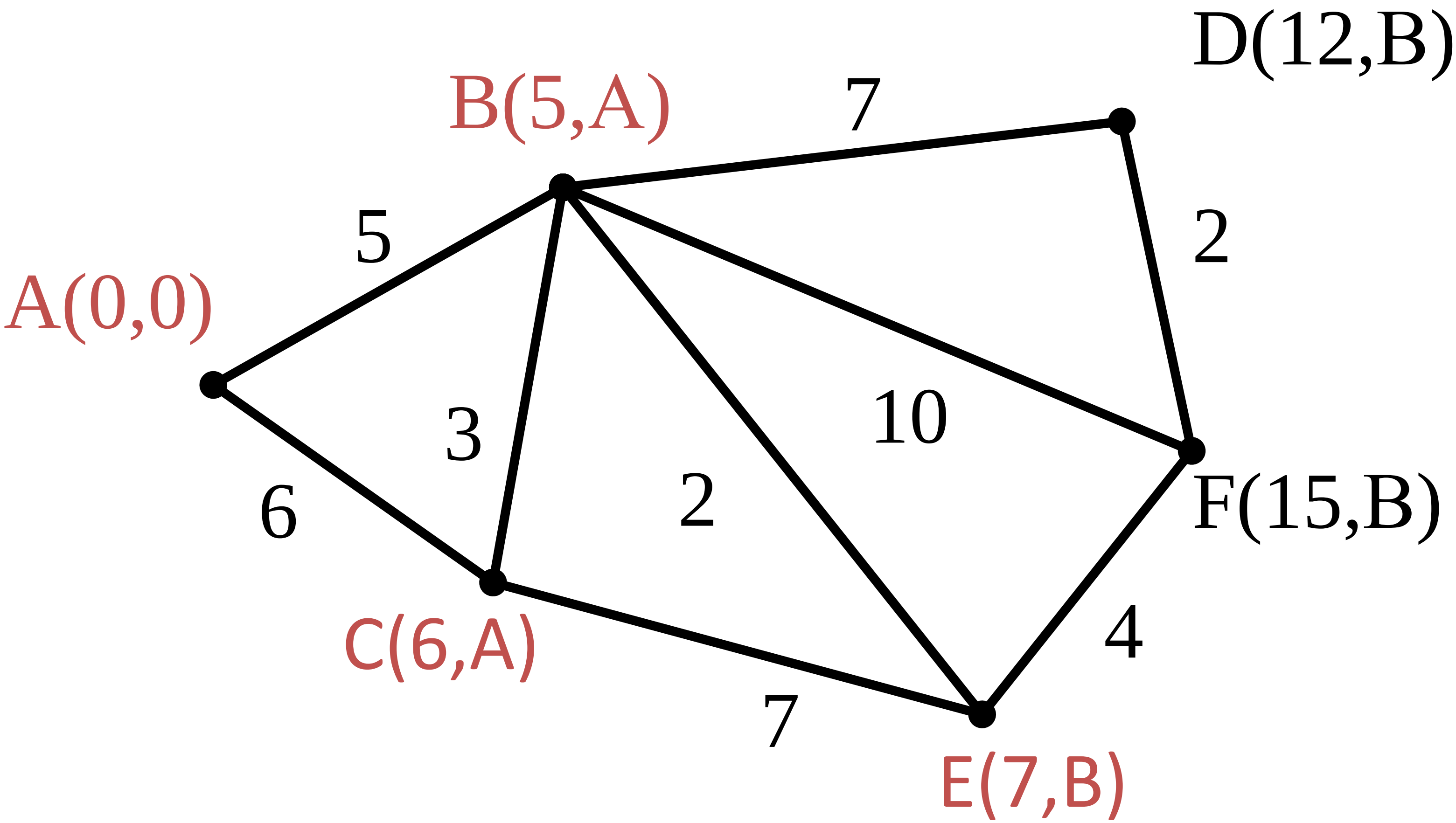
Изо всех текущих меток расстояний $\min\{12,15,7,6\} = 6 \Rightarrow$ вершину C(6,A) делаем постоянной



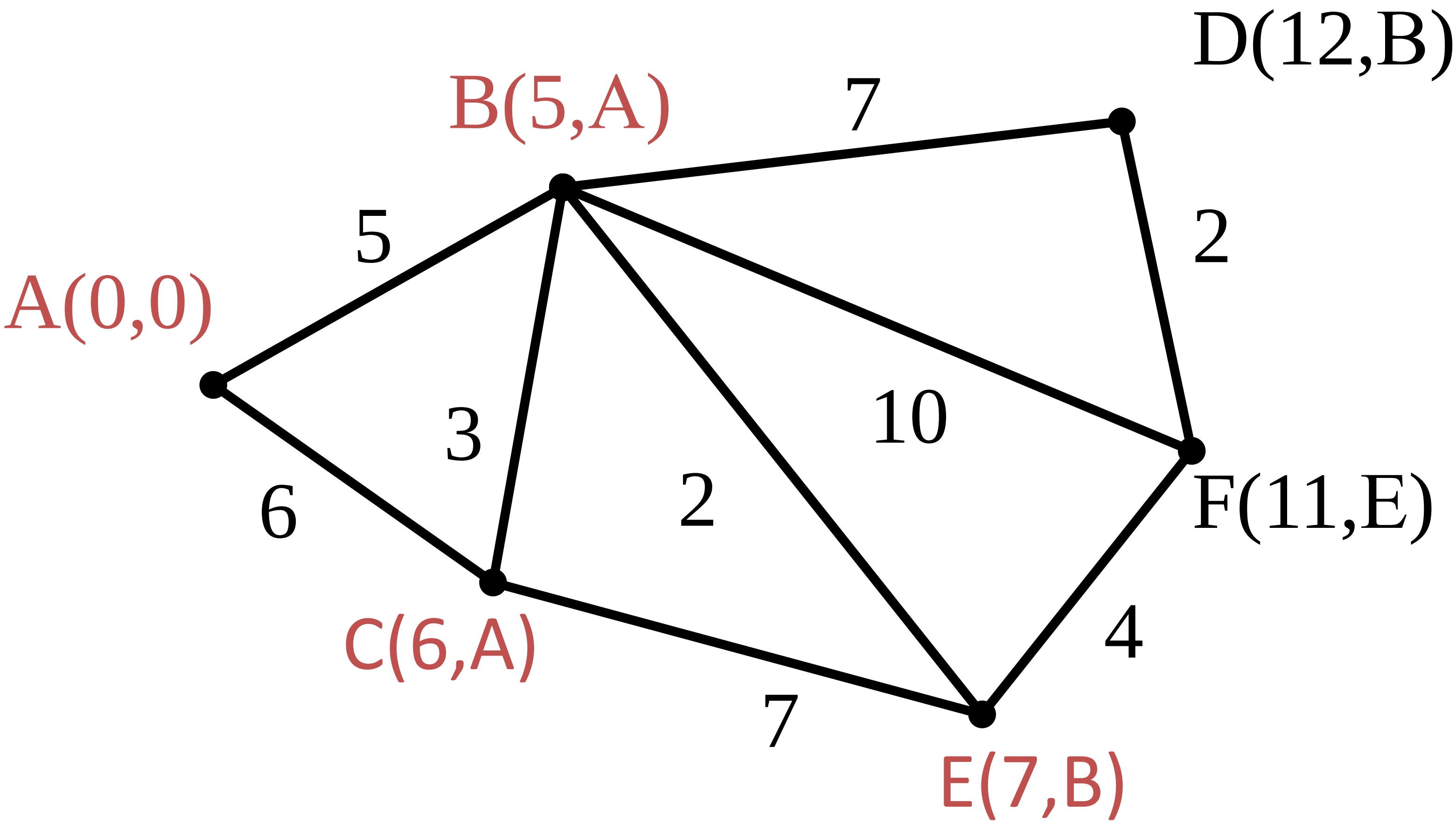
Пересчитываем расстояния от C до смежных с ней временных вершин



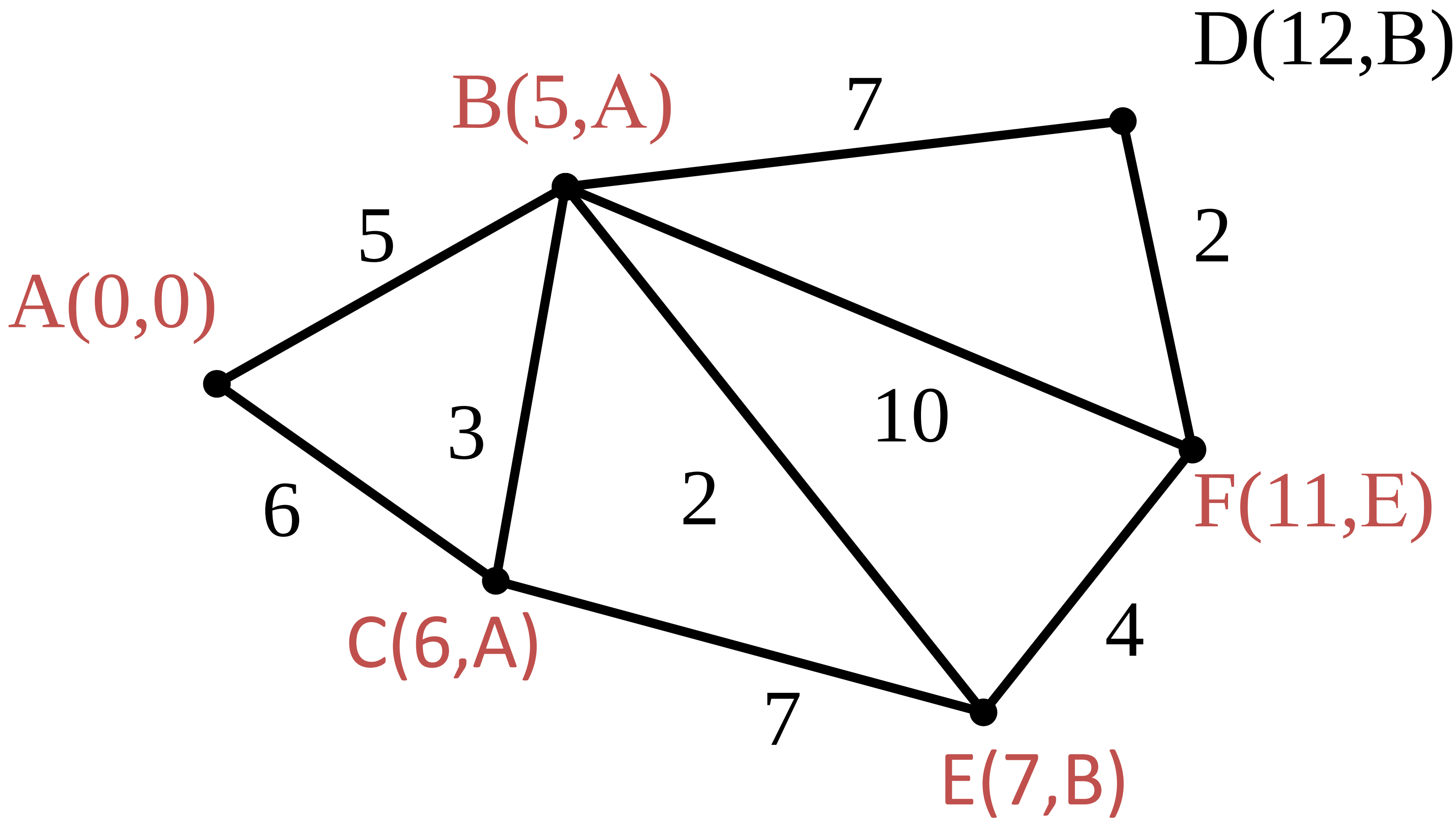
Вершину с наименьшим расстоянием делаем постоянной – E.



Пересчитываем расстояния для временных вершин, смежных с E



Выбираем вершину с наименьшим расстоянием и делаем её постоянной (F)



Оформление в виде таблицы:

	Пройденные вершины	w	d(w)	d(B)	d(C)	d(D)	d(E)	d(F)
0	A	A	0	5	6	∞	∞	∞
1	A,B	B	5	-	6	12	7	15
2	A,B,C	C	6	-	-	12	7	15
3	A,B,C,E	E	7	-	-	12	-	11
4	A,B,C,E,F	F	11	-	-	12	-	-