Воротницкий Ю.И.

Исследование операций

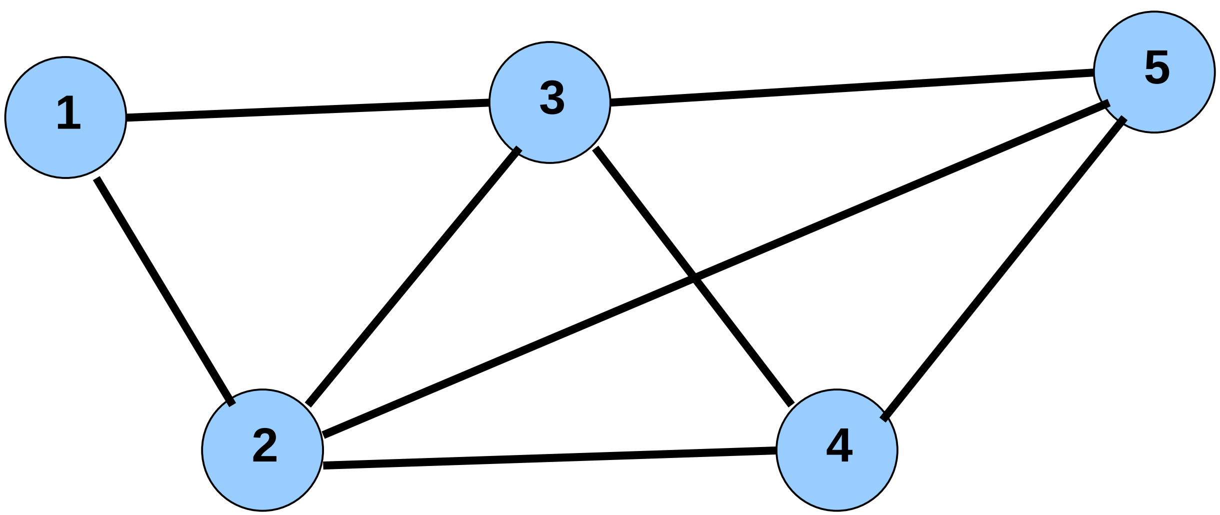
Сетевые модели

Сетевые модели

Основные элементы сетевых моделей

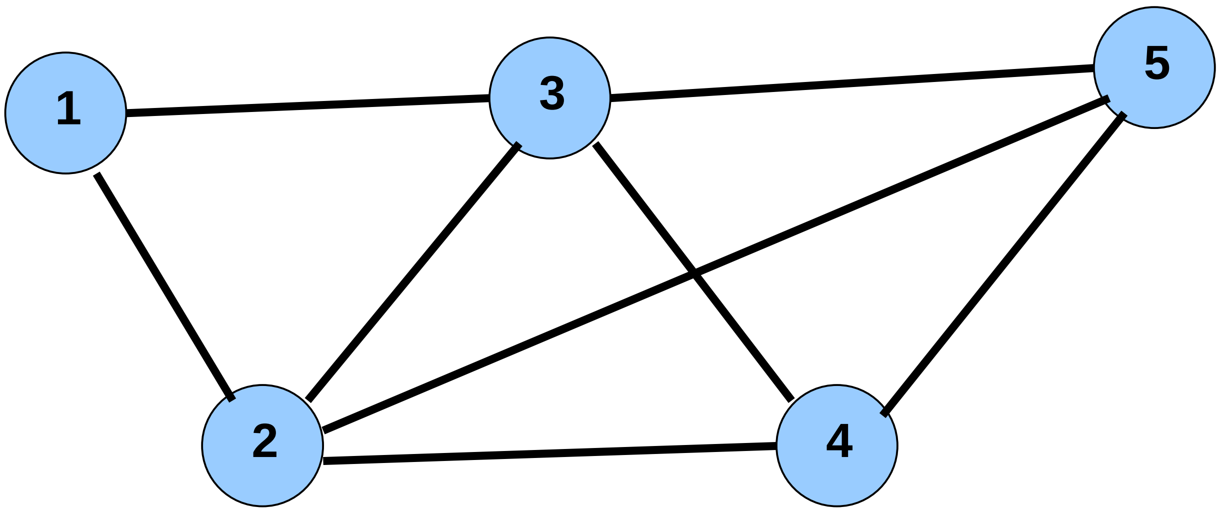
Сетевые модели.
Основные определения

- **Сеть** состоит из множества узлов (вершин), связанных дугами или ребрами.
- Сеть описывается парой множеств (N,A), где N – множество узлов, а A – множество ребер.



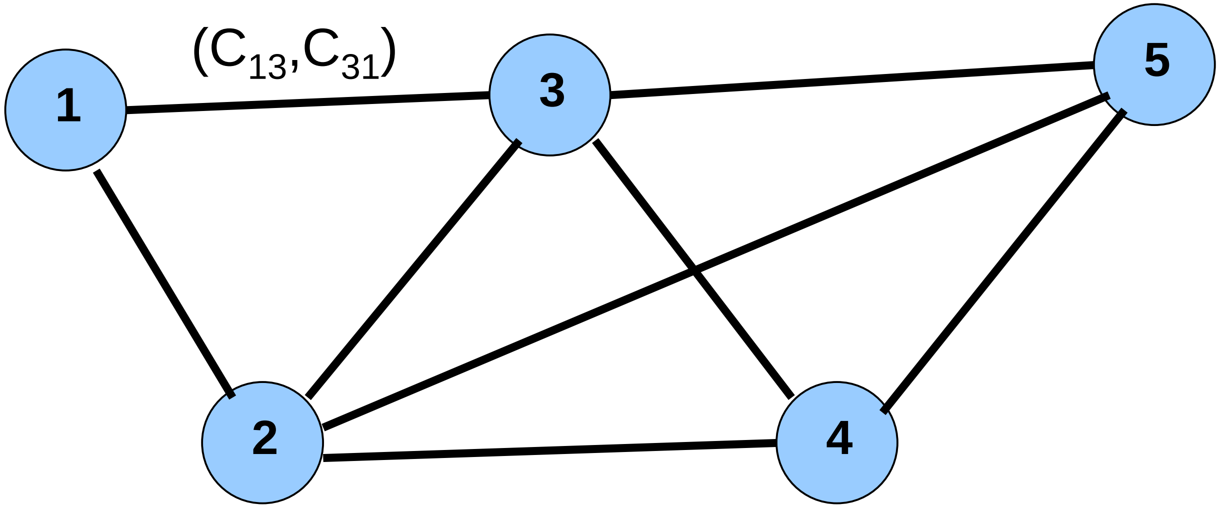
Сетевые модели.
Основные определения

- Показанная на рисунке сеть описывается следующим образом:
  - N={1,2,3,4,5}
  - A={(1,3), (1,2), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5)}



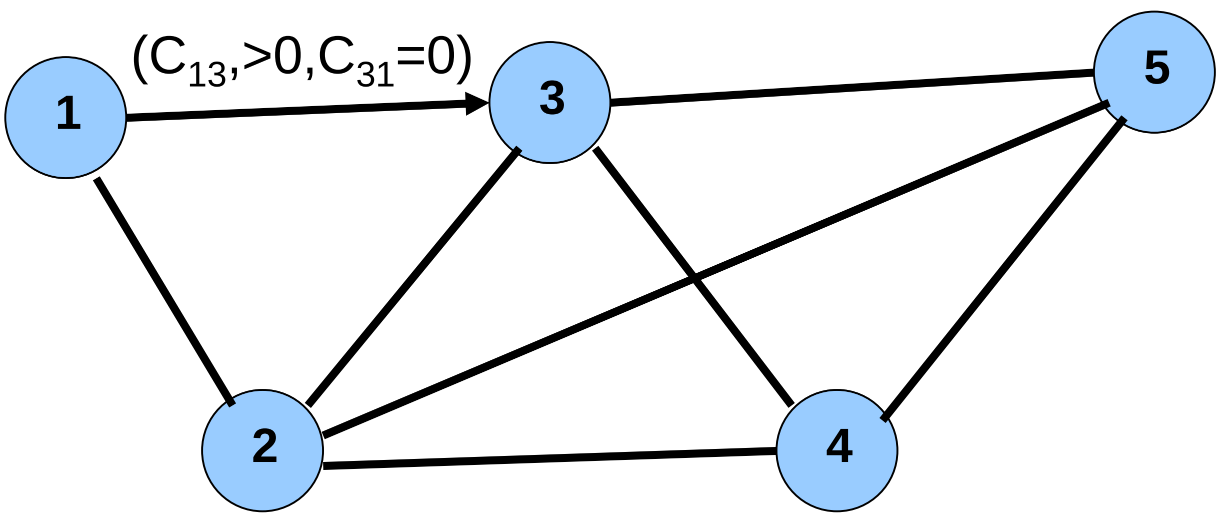
Сетевые модели.
Основные определения

- С каждым типом сети связан определенный тип **потоков**.
- В общем случае потоки в сети ограничены **пропускной способностью** ее ребер C_{ij}, которая в общем случае может быть как конечной, так и бесконечной (максимальный ток в цепи электропитания, пропускная способность телекоммуникационных каналов и т.д.).



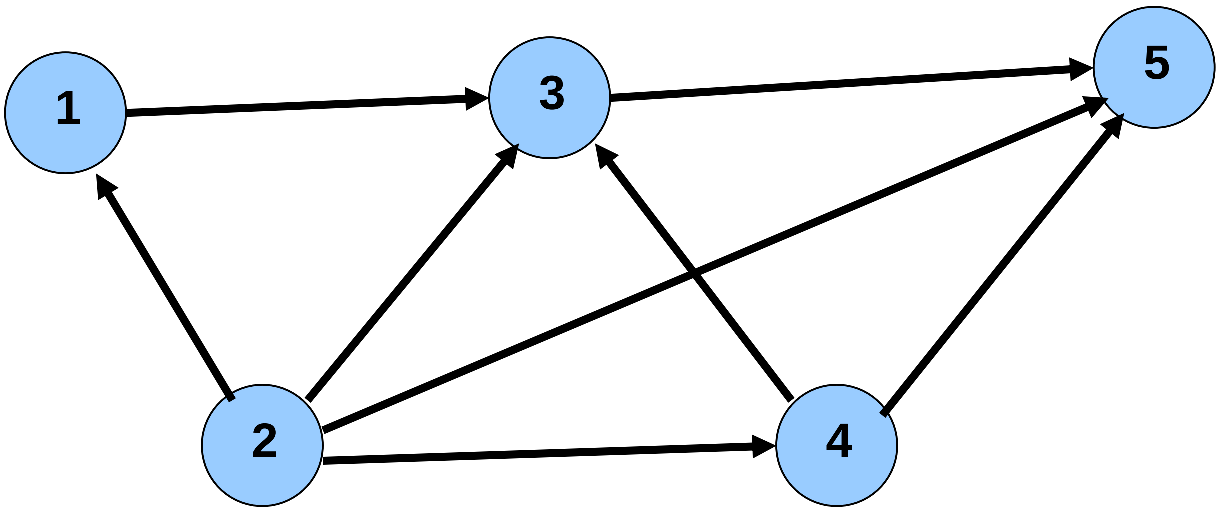
Сетевые модели.
Основные определения

- Ребро называется **направленным** (ориентированным), если в одном направлении возможен только положительный поток, а в противоположном – только нулевой
- В этом случае ребро называют **дугой**



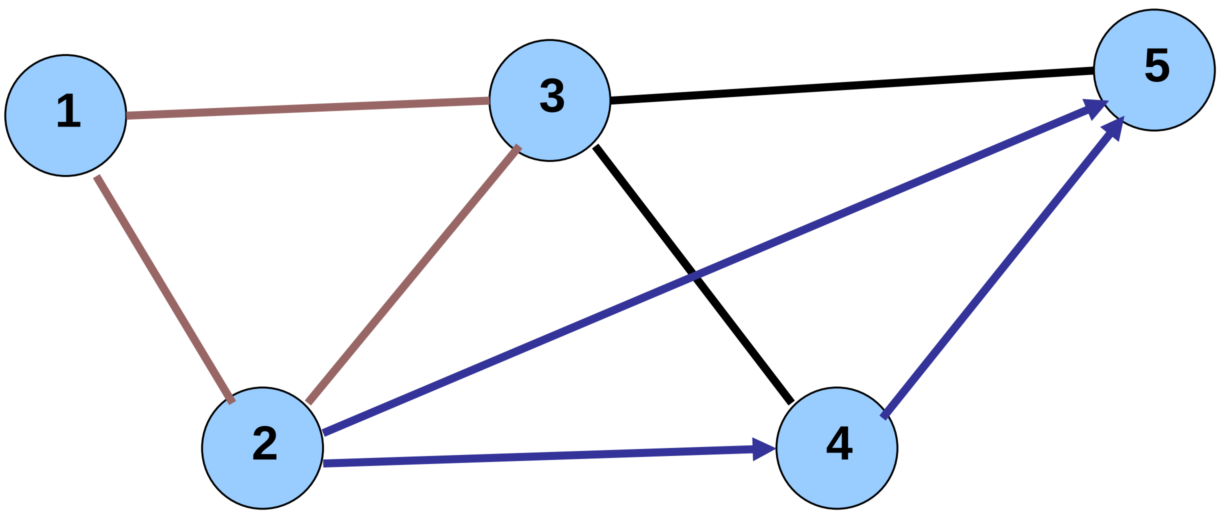
Сетевые модели.
Основные определения

- В **ориентированной сети** все ребра ориентированы:
  - N={1,2,3,4,5}
  - A={(1,3), (2,1), (2,3), (2,4), (2,5), (4,3), (3,5), (4,5)}



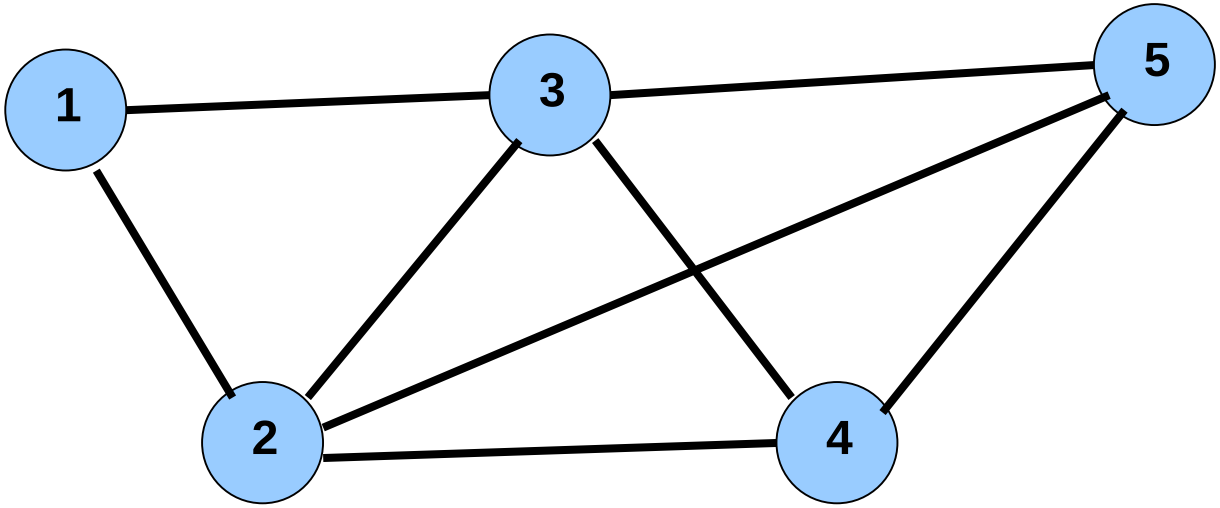
Сетевые модели.
Основные определения

- **Путем** называется последовательность различных ребер, соединяющих два узла, независимо от направления потока в каждом ребре.
- Путь формирует **цикл**, если начальный и конечный узлы совпадают.
- **Ориентированный цикл** – цикл, в котором дуги ориентированы в определенном направлении.



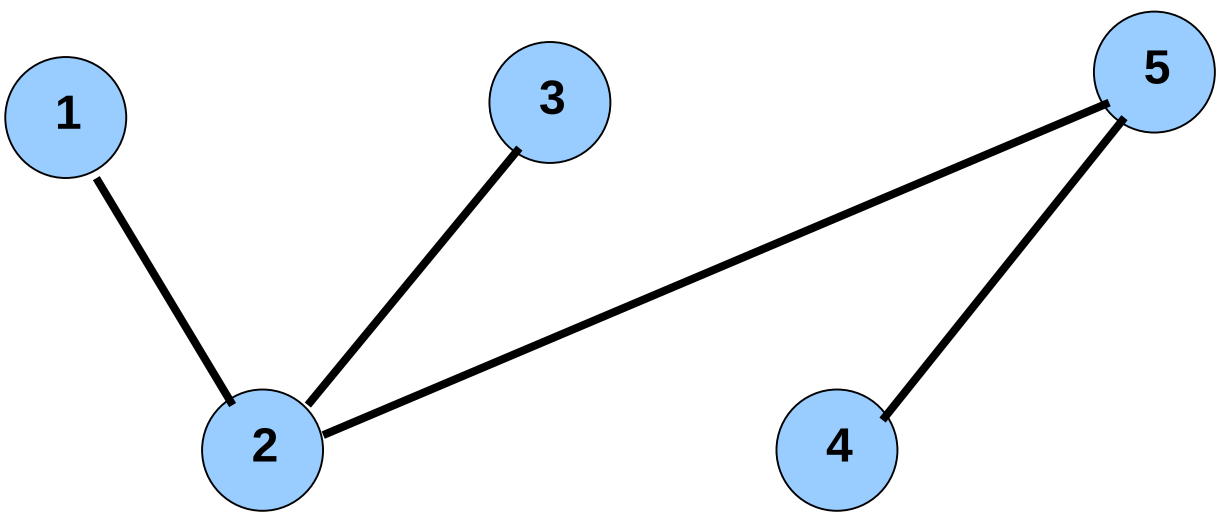
Сетевые модели.
Основные определения

- **Связная сеть** – такая сеть, у которой любые два узла связаны по крайней мере одним путем.
- Это – связная сеть:
  - N={1,2,3,4,5}
  - A={(1,3), (1,2), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5)}



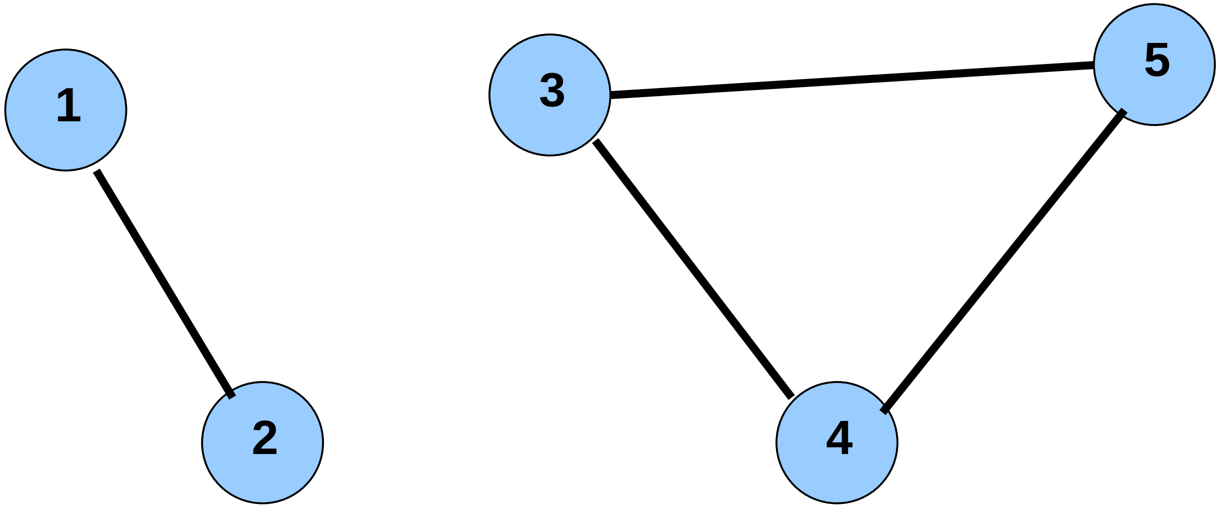
Сетевые модели.
Основные определения

- **Связная сеть** – такая сеть, у которой любые два узла связаны по крайней мере одним путем.
- Это – тоже связная сеть:
  - N={1,2,3,4,5}
  - A={(1,2), (2,3), (2,5), (4,5)}



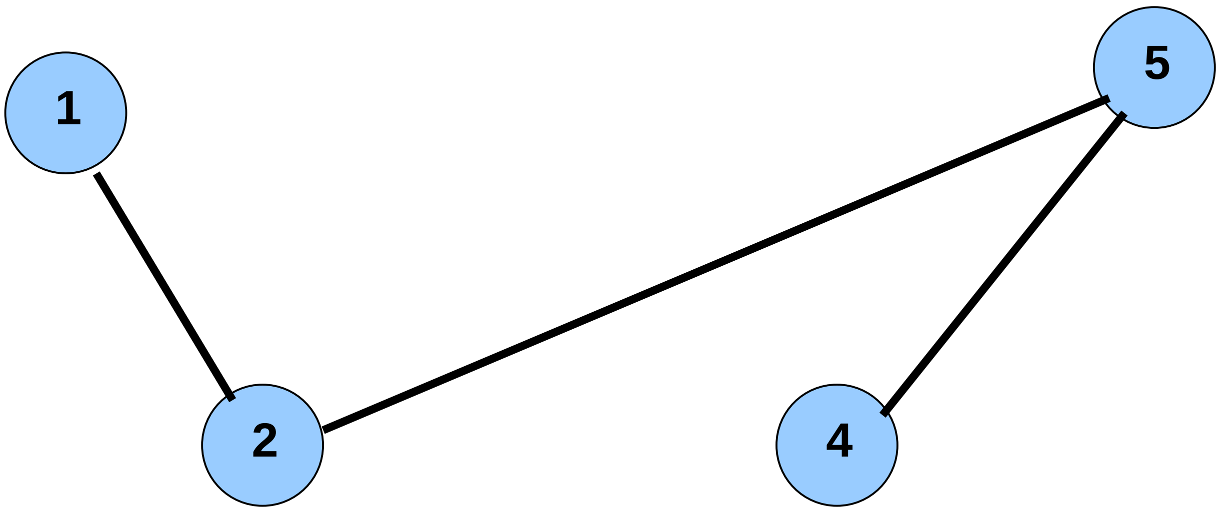
Сетевые модели.
Основные определения

- **Связная сеть** – такая сеть, у которой любые два узла связаны по крайней мере одним путем.
- А эта сеть связной не является:
  - N={1,2,3,4,5}
  - A={(1,2), (3,4), (3,5), (4,5)}



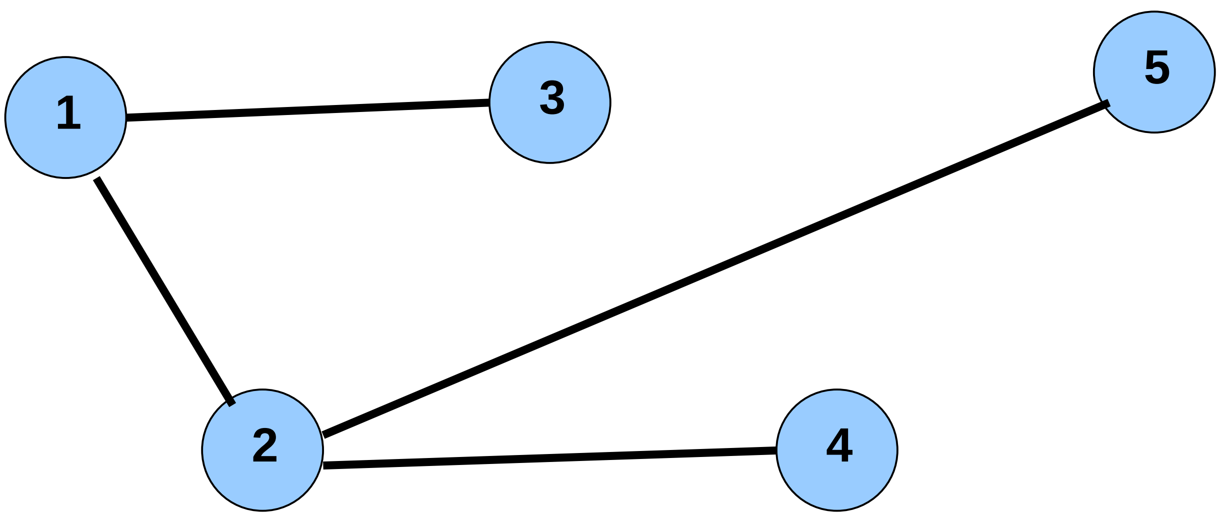
Сетевые модели.
Основные определения

- **Деревом** называется связная сеть, содержащая подмножество узлов исходной сети и не имеющая циклов.
- Пример дерева:
  - N={1,2,4,5}
  - A={(1,2), (2,5), (4,5)}



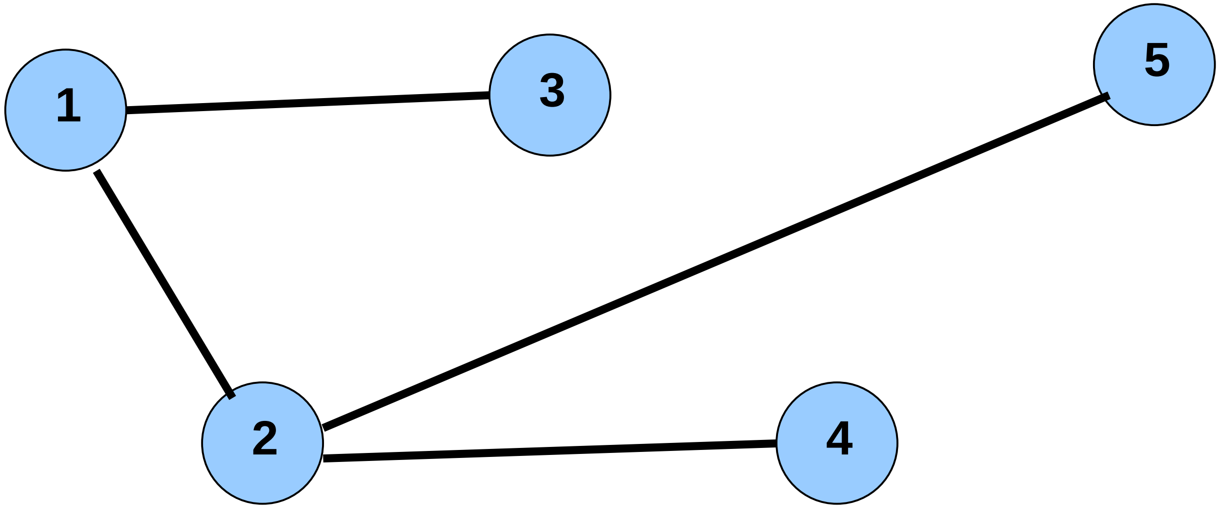
Сетевые модели.
Основные определения

- **Остовное дерево –** дерево, содержащее все узлы сети.
- Пример остовного дерева:
  - N={1,2,3,4,5}
  - A={(1,3), (1,2), (2,4), (2,5)}



Сетевые модели.
Представление сетей

- Сеть G=(N,A) может быть полностью определена **простым перечислением множеств** N и A.
  - N={1,2,3,4,5}
  - A={(1,3), (1,2), (2,4), (2,5)}
- Такой способ не позволяет легко анализировать свойства сетей

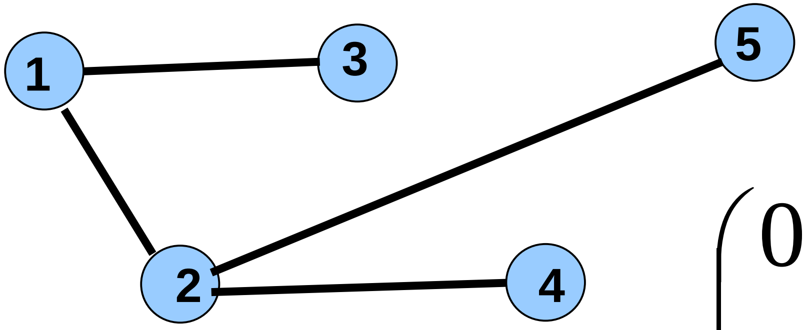


Сетевые модели.
Представление сетей

- **Матрица смежности.**
  - Любая сеть G=(N,A) с m узлами (вершинами) может быть представлена матрицей $A(G)=[a_{ij}]$ размера m x m.
  - Для этого узлы должны быть перенумерованы (или помечены метками порядкового типа v_1, v_2, …, v_m).
  - $a_{ij}=1$, если v_i смежен с v_j, в противном случае $a_{ij}=0$.
  - Для неориентированной сети G A(G) всегда будет симметричной матрицей (0,1) с нулями на диагонали.
  - Для ориентированной матрицы G только один из элементов a_{ij}, a_{ji} может отличаться от нуля.
  - При необходимости значения 0 и 1 в матрице можно заменить пропускными способностями или стоимостями путей.

Сетевые модели.

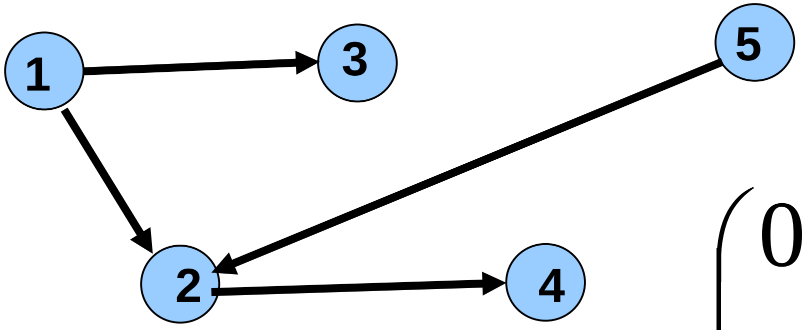
Представление сетей Матрица смежности



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Сетевые модели.

Представление сетей Матрица смежности



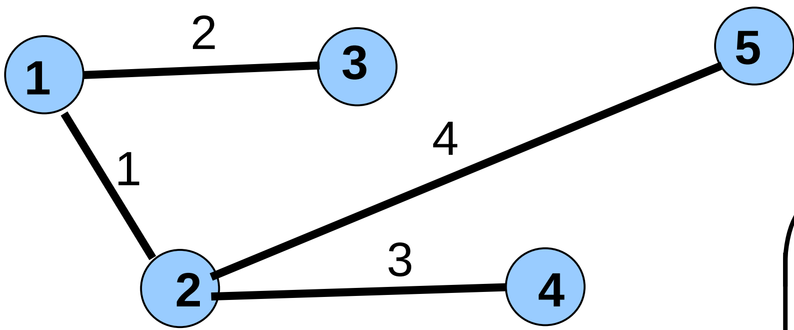
$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Сетевые модели.
Представление сетей

- **Матрица инцидентности.**
  - Любая сеть G=(N,A) с m узлами и n ребрами может быть представлена матрицей $I(G)=[b_{ij}]$ размера m x n.
  - Для этого узлы должны быть перенумерованы (или помечены метками порядкового типа $v_1, v_2, \ldots, v_m$), а ребра также перенумерованы (или помечены метками порядкового типа $e_1, e_2, \ldots, e_n$), .
  - $b_{ij}=1$, если v_i инцидентен e_j, в противном случае $b_{ij}=0$.
  - Каждый j-й столбец матрицы I(G), соответствующий j-му ребру, всегда содержит ровно две единицы.
  - Никакие два столбца не могут быть идентичны.
  - Матрица инцидентности полезна для решения сетевых задач, касающихся анализа циклов.

Сетевые модели.

Представление сетей Матрица инцидентности



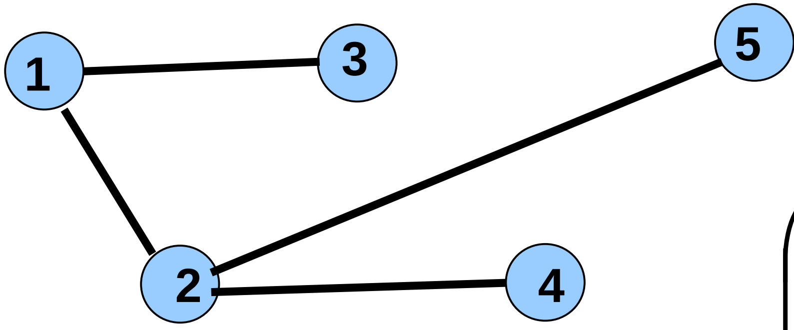
$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Сетевые модели.
Представление сетей

- **Векторы смежности.**
  - Любая сеть G=(N,A) с m узлами может быть представлена матрицей $C(G)=[c_{ij}]$ размера m x m-1.
  - Для этого узлы должны быть перенумерованы (или помечены метками порядкового типа v_1, v_2, …, v_m).
  - Каждая i-я строка матрицы соответствует i-му узлу.
  - Значения элементов i-й строки c_{ij} – номера узлов, смежных с v_i.
  - Каждая i-я строка представляет собой вектор смежности для i-го узла сети.
  - Порядок элементов в векторе смежности в общем случае произволен.
  - Векторы смежности целесообразно использовать, когда задача решается за небольшое число просмотров каждого ребра в G.

Сетевые модели.

Представление сетей Вектор смежности

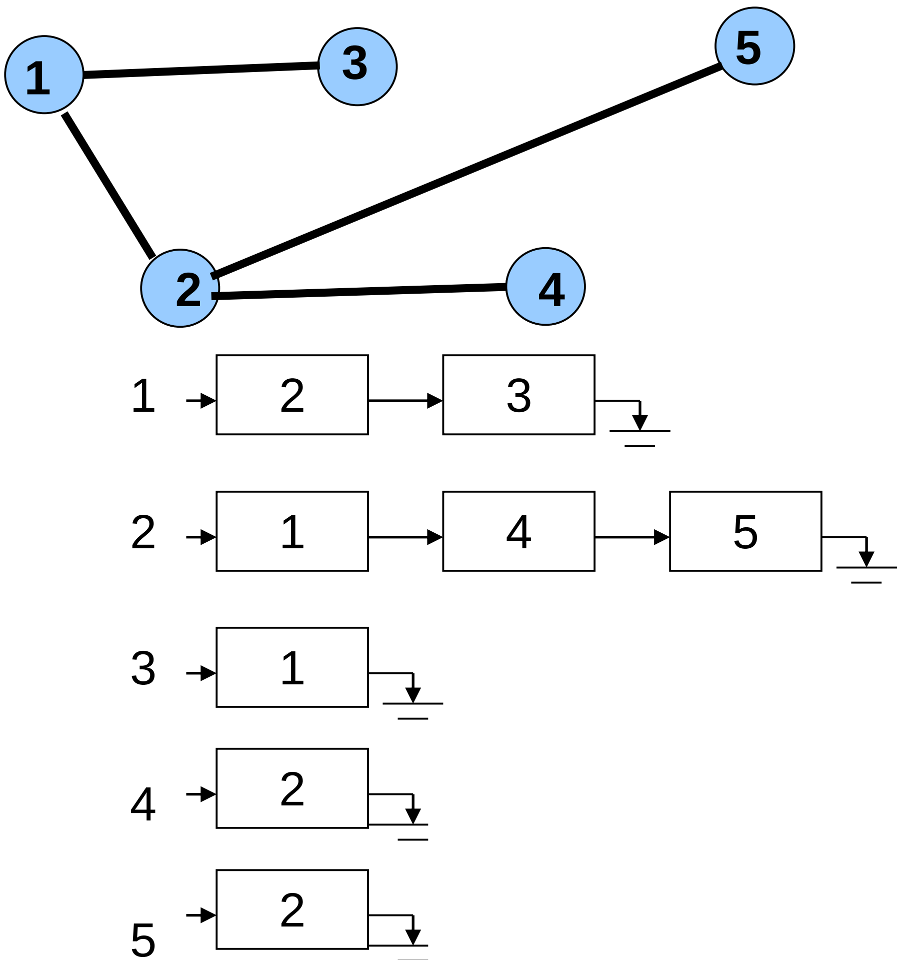


$$\begin{pmatrix} 2 & 3 & 0 & 0 \\ 5 & 4 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$$

Сетевые модели.
Представление сетей

- **Списки смежности.**
  - Списки смежности – один из наиболее эффективных способов представления сети G=(N,A), где N – m вершин, A – n ребер.
  - В этом случае сеть представляется с помощью m списков, каждый из которых может иметь от 0 до m-1 элемента.
  - Информационное поле каждого элемента i-го списка содержит номер вершины, смежной с i-й
  - Удобно использовать одномерный массив из m элементов, причем каждый элемент массива представляет собой линейный список, представляющий ненулевые компоненты вектора смежности для соответствующей вершины

Сетевые модели.
Представление сетей Списки смежности



Сетевые модели.
Постановки задач и базовые алгоритмы

Сетевые задачи и алгоритмы

- Построение сети минимальной длины (стоимости)
  - Алгоритм нахождения минимального остовного дерева
- Нахождение кратчайшего маршрута (маршрута минимальной стоимости)
  - Алгоритмы нахождения кратчайшего пути (Дейкстры, Флойда)
- Сетевое планирование
  - Алгоритм нахождения критического пути
- Определение максимальной пропускной способности между узлами
  - Алгоритм определения максимального потока
- Нахождение потока наименьшей стоимости
  - Алгоритм минимизации стоимости потока в сети с ограниченной пропускной способностью

Сетевые модели.
Постановки задач и базовые алгоритмы

- Все перечисленные задачи можно сформулировать и решить как задачи линейного программирования.
- Этот подход неэффективен, так как специфическая структура этих задач позволяет построить для них специальные, более эффективные, вычислительные алгоритмы

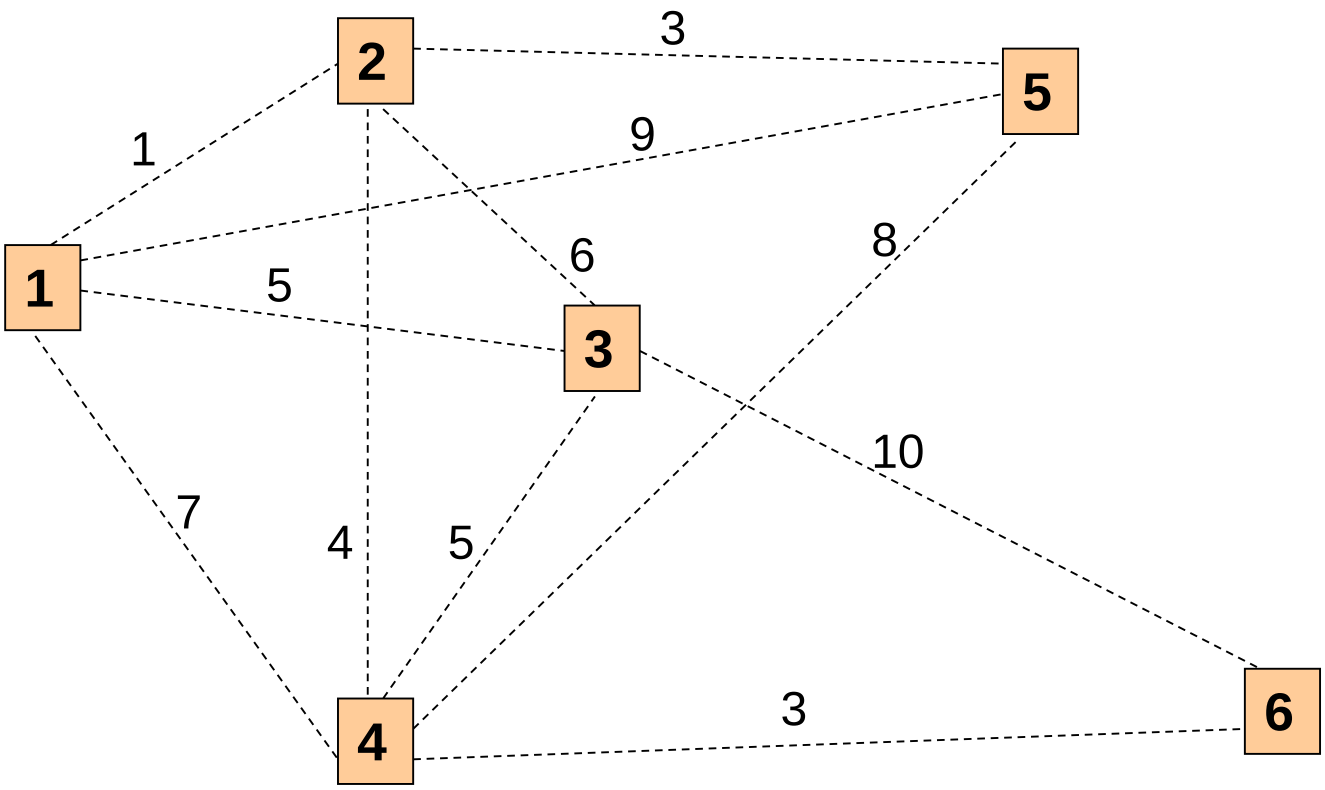
Построение минимального остовного дерева

Сетевые модели.
Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети

- Банк имеет в городе 6 крупных отделений.
- С целью создания корпоративной информационной системы необходимо связать их с помощью опорной оптоволоконной сети минимальной стоимости, полагая, что она определяется общей длиной коммуникаций.
- Расстояния между офисами известны.
- Разумеется, такая постановка задачи в значительной степени идеализирована:
  - не учитываются параметры и стоимость коммуникационного оборудования в узлах сети, а также пропускная способность каналов;
  - не учитывается, что прокладка по существующим телефонным канализациям дешевле, однако при этом могут существенно увеличиваться длины кабелей;
  - не рассматривается необходимость резервирования каналов связи.

Сетевые модели.

Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети



Сетевые модели.
Алгоритм построения минимального остовного дерева

- Заданы N={1,2,…m} – множество узлов сети. Заданы длины (стоимости) возможных дуг s_{ij}, которые можно провести между узлами i и j.
- Необходимо соединить все узлы сети с помощью путей (дуг) наименьшей суммарной длины (стоимости).
- Очевидно, что для этого необходимо построить дерево, связвающее все узлы с помощью дуг наименьшей общей длины (стоимости).

- Обозначим C_k – множество узлов, соединенных алгоритмом после выполнения k-й итерации, D_k – множество узлов сети, не соединенных с узлами множества C_k после выполнения k-й итерации алгоритма.

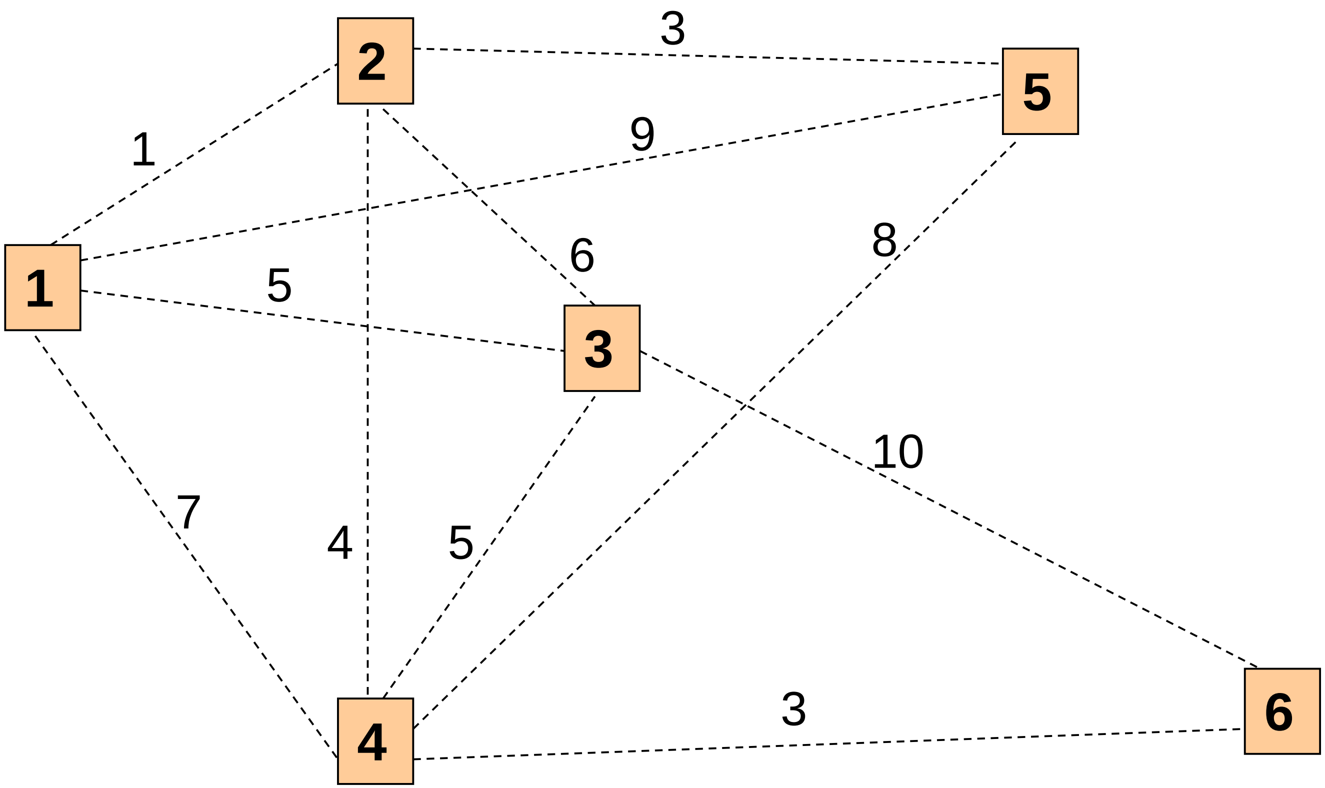
4. Сетевые модели.

4.4. **Алгоритм построения минимального остовного дерева**

- **Шаг 0.** Положить C_0 = (пустое множество), D_0=N.
- **Шаг 1.** Выбрать любой узел i из множества D_0 и определить C_1 ={i}, D_1=N-{i}. Положить k=2.
- **Шаг 2.** Пока множество D_0 не является пустым, выполнить:
  - **Шаг 2.1.** В множестве D_{k-1} выбрать узел j, который соединен самой короткой дугой с каким-либо узлом из C_{k-1}.
  - **Шаг 2.2.** $C_k=C_{k-1}+\{j\}$, $D_k=D_{k-1}-\{j\}$.
  - **Шаг 2.3.** k=k+1.
- **Шаг 3.** Завершить работу.

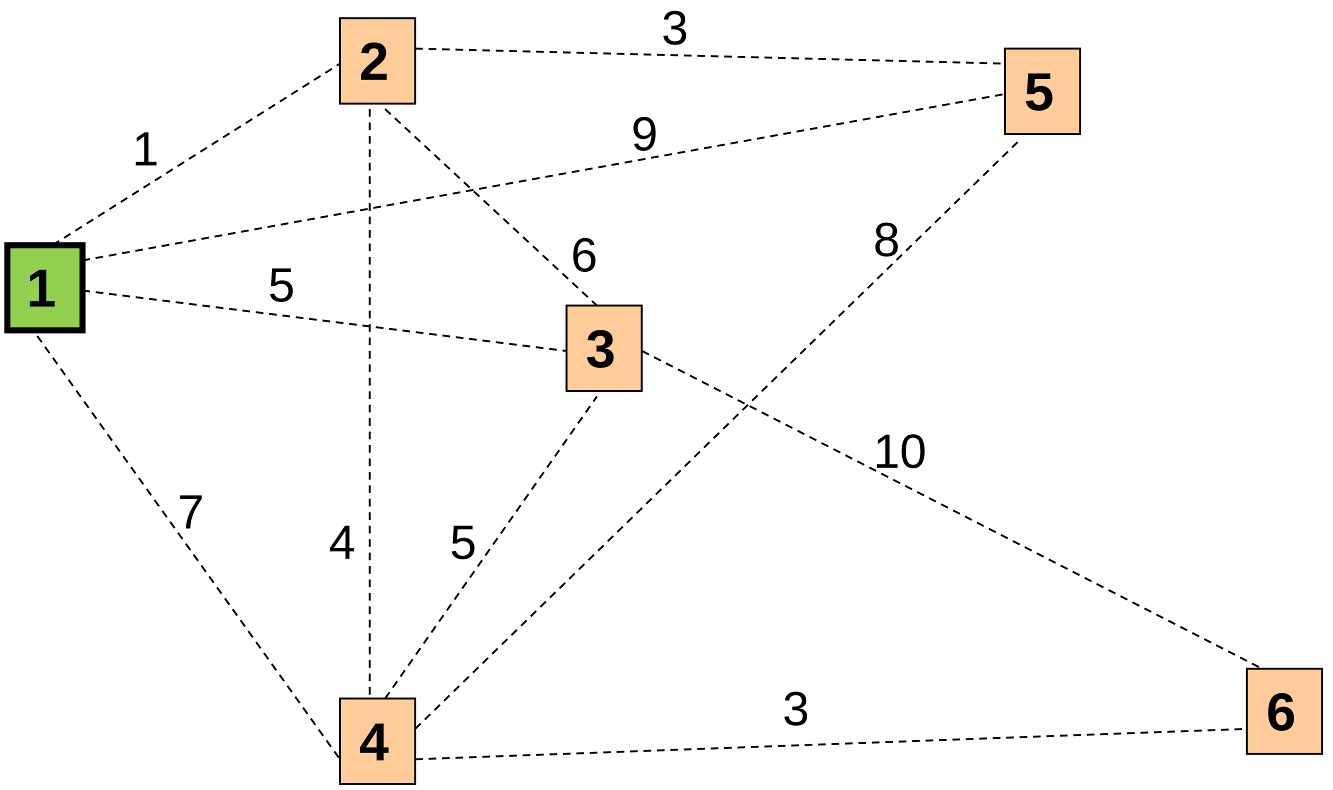
Сетевые модели.

Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети



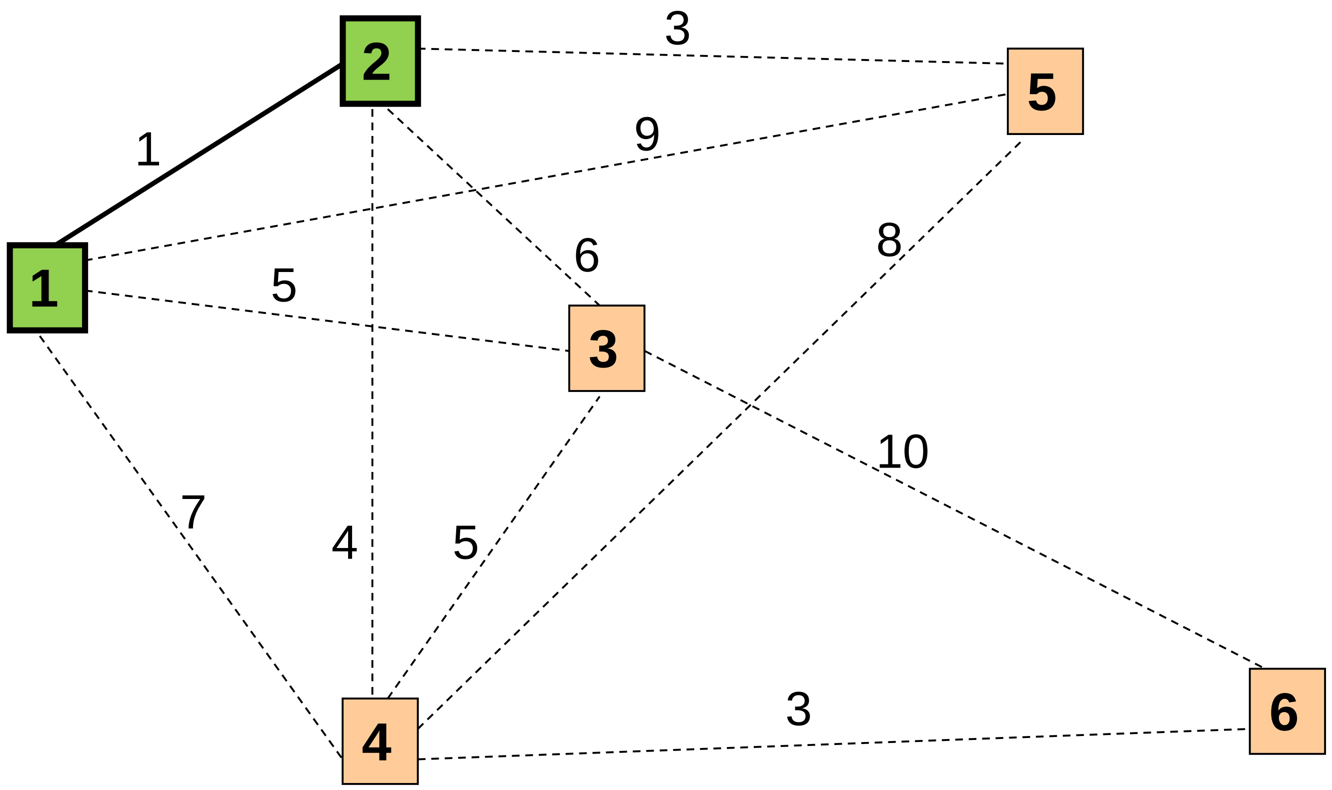
Сетевые модели.

Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети



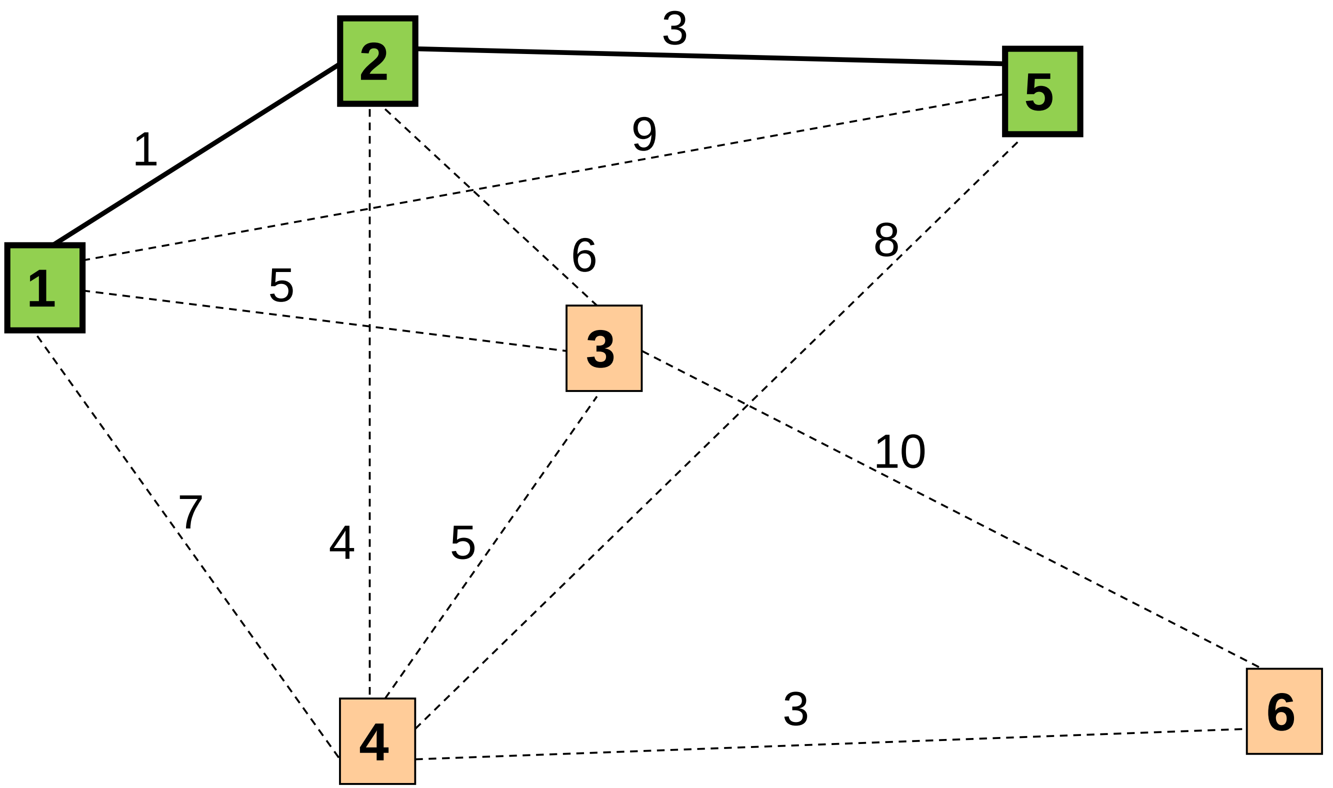
Сетевые модели.

Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети



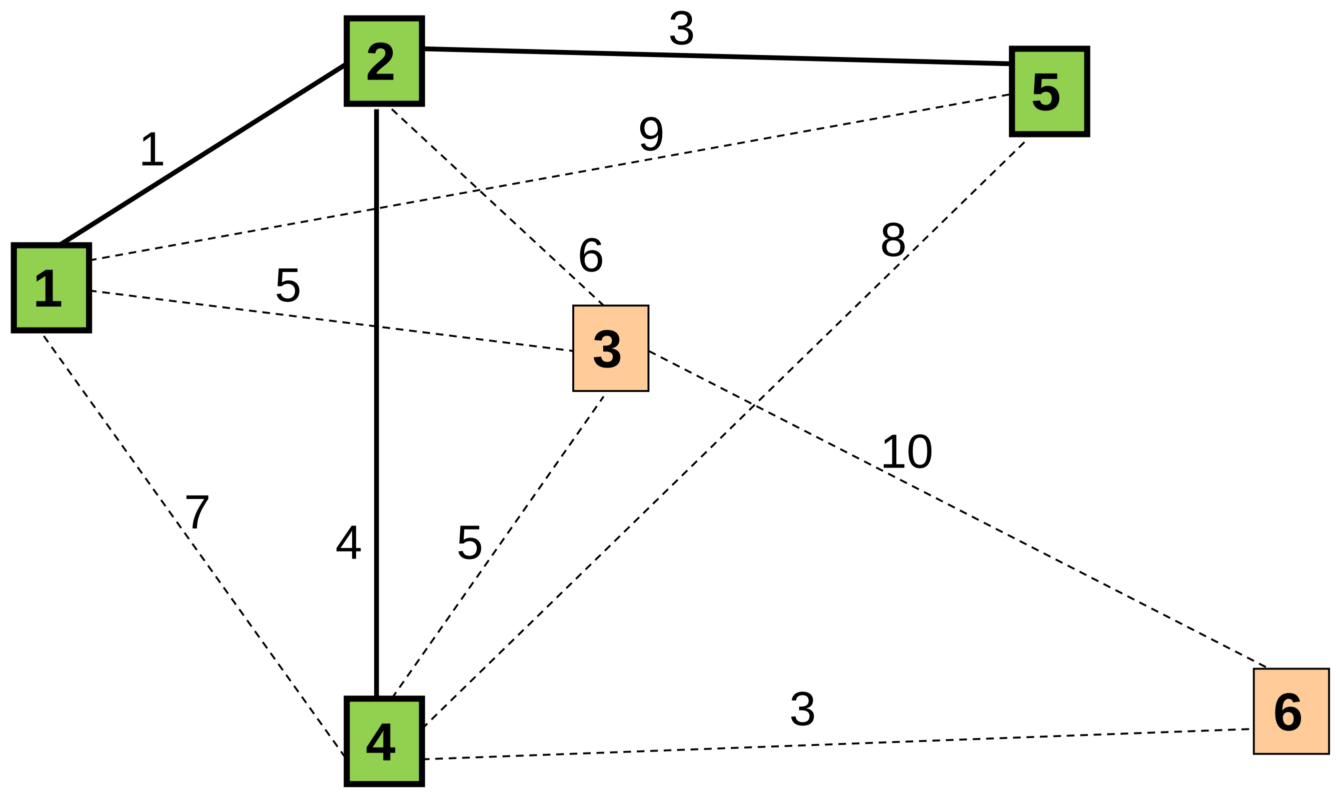
Сетевые модели.

Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети



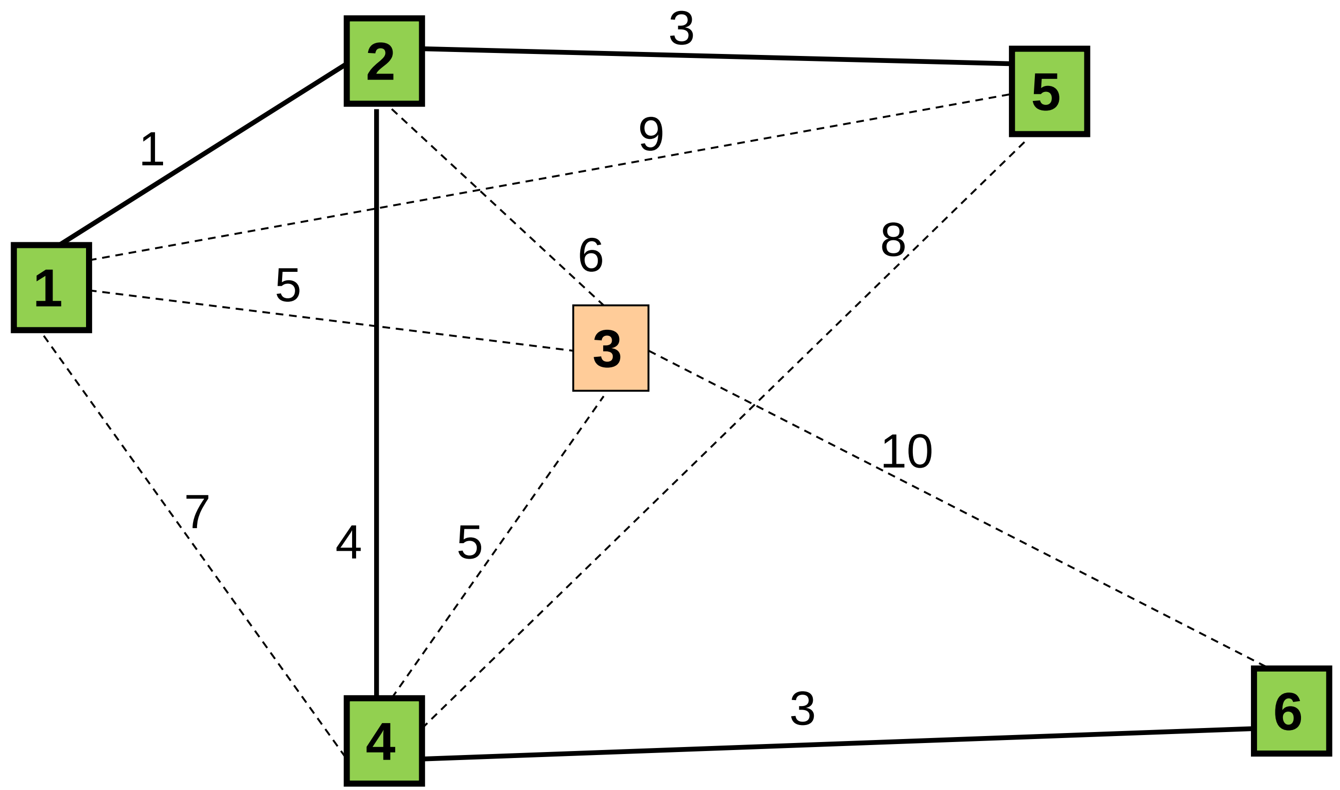
Сетевые модели.

Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети



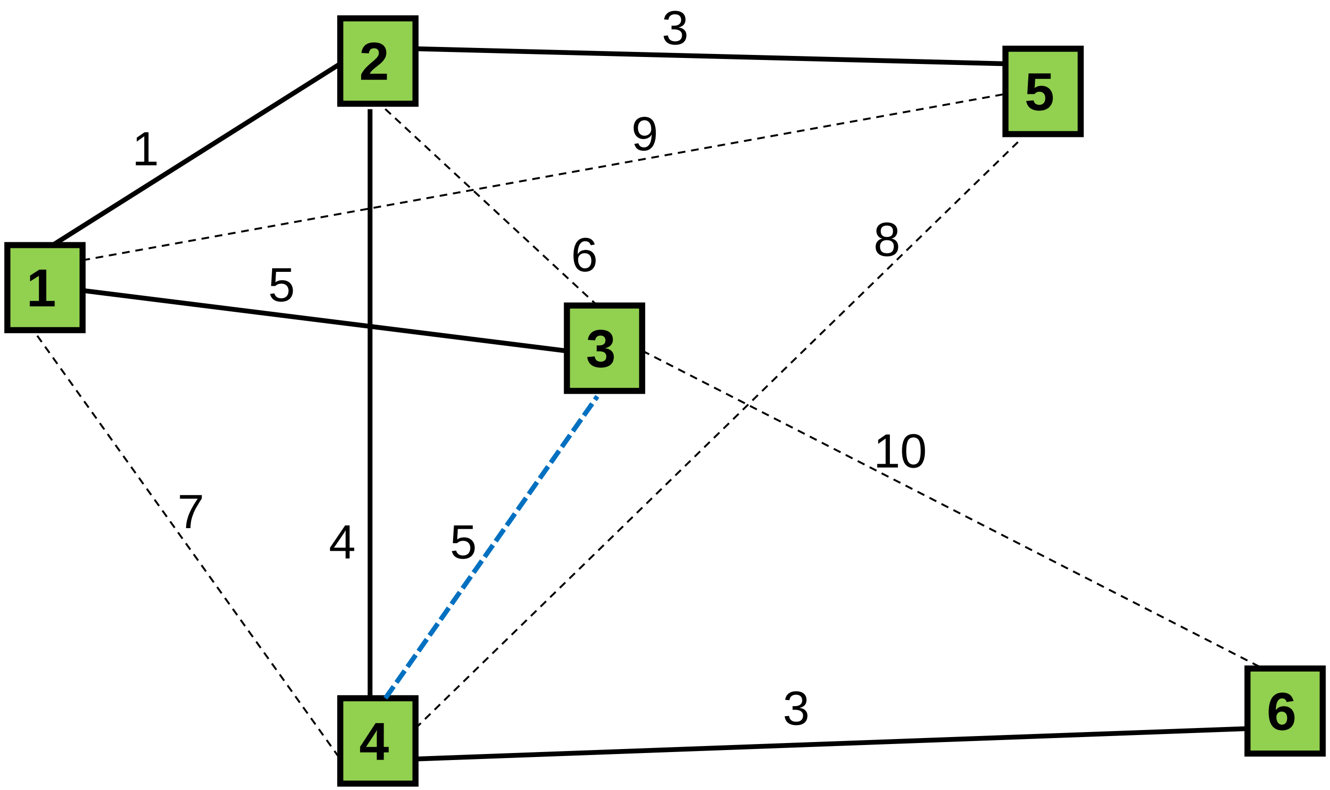
Сетевые модели.

Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети



Сетевые модели.

Алгоритм построения минимального остовного дерева. Задача построения опорной телекоммуникационной сети



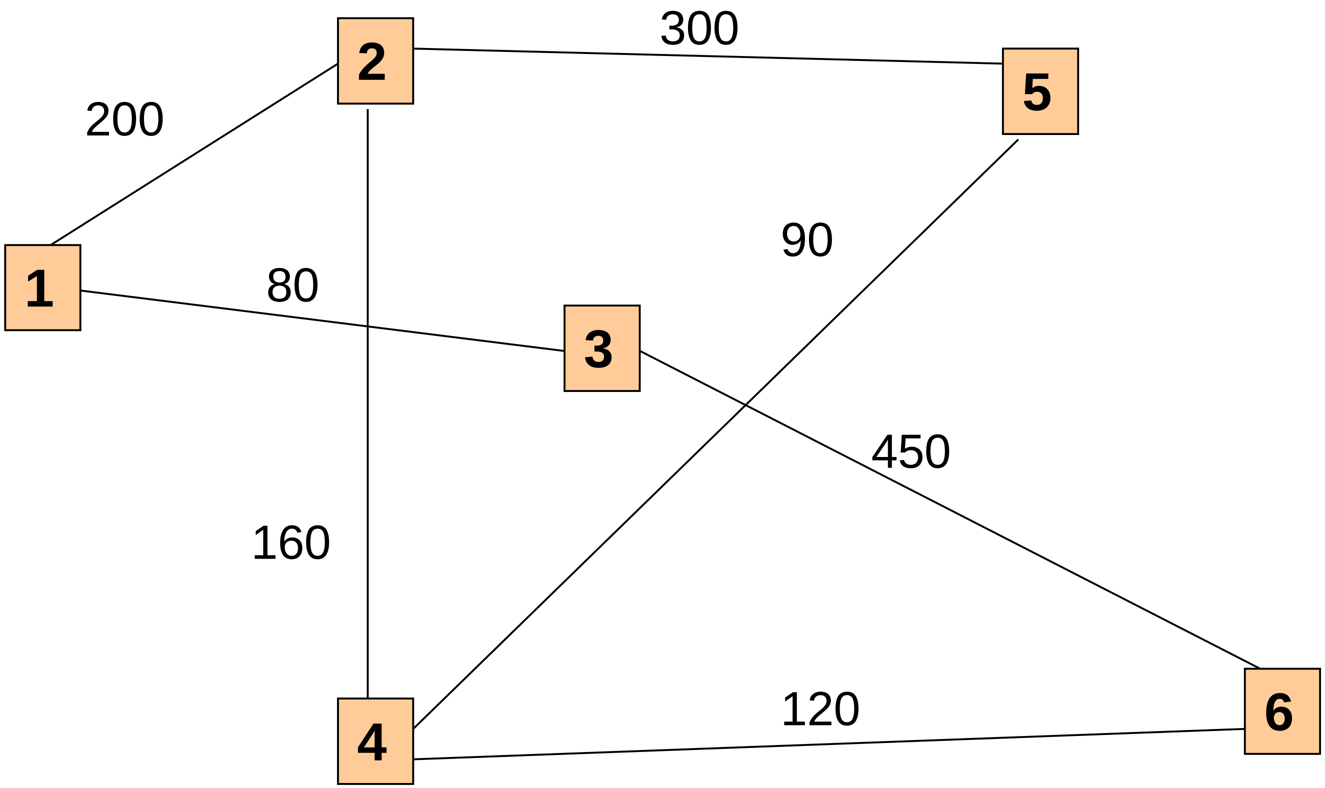
Нахождение кратчайшего и критического путей

Сетевые модели.
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.

- Банк имеет в городе 6 крупных отделений, соединенных оптоволоконными линиями передачи.
- Необходимо организовать видеоконференцсвязь между центральным офисом (узел 1) и отделениями. Для этого необходимо предложить схему статической маршрутизации пакетов, минимизирующую времена задержек от узла 1 до каждого из остальных узлов. Средние времена задержек при передаче от узла к узлу в условиях нормальной загрузки сети известны.
- Разумеется, эта задача тоже идеализирована. В частности:
  - не учитывается влияние изменения самого трафика видеоконференций на средние времена задержек;
  - не учитываются другие параметры, влияющие на качество видеоконференцсвязи (вариации задержек, вероятность потери пакетов).

Сетевые модели.

Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Сетевые модели.

Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.

- **Алгоритм Дейкстры.**
- При переходе от узла i к следующему узлу j используется специальная процедура пометки ребер.
- Обозначим через u_i кратчайшее расстояние от исходного узла 1 до узла i, через d_{ij} – длину ребра (i,j). Тогда для узла j определим метку $[u_j,i]$ следующим образом: $[u_j,i] = [u_i+ d_{ij},i]$.
- Метки могут быть двух типов: временные и постоянные.
- Временная метка может быть заменена на другую временную, если будет найден более короткий путь к данному узлу.
- Статус временной метки заменяется на постоянный, когда станет очевидным, что не существует более короткого пути от исходного узла к данному.

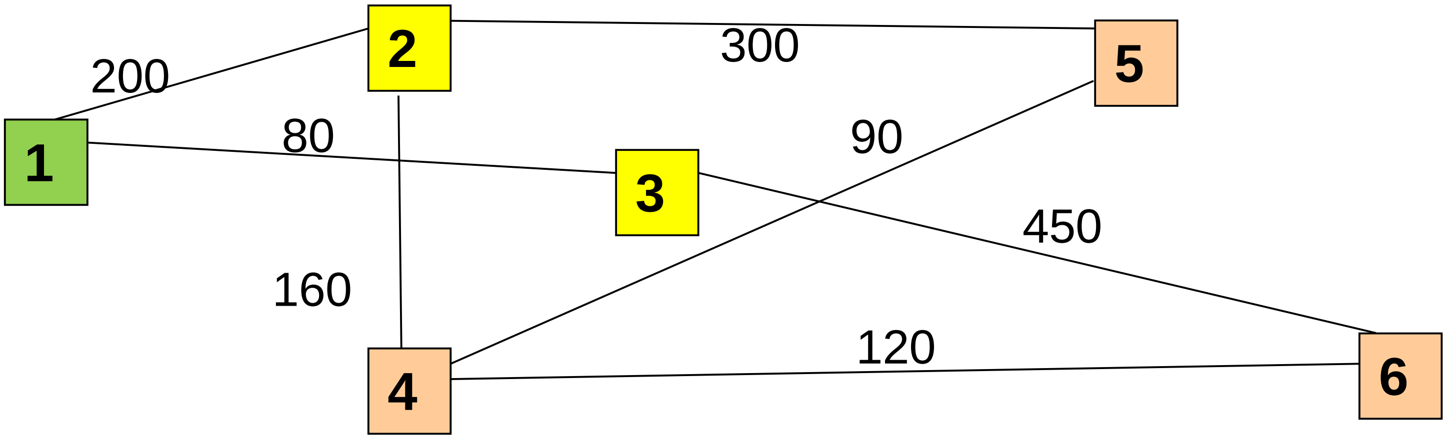
Сетевые модели.

Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.

- **Алгоритм Дейкстры.**
- Шаг 0. Исходному узлу (узел 1) присваивается метка [0,-]. Положить i=1.
- Шаг i.
  - Вычислить временные метки $[u_i + d_{ij}, i]$ для всех узлов j, которые можно достичь из узла i и *которые не имеют постоянных меток*. Если узел j уже имеет временную метку, полученную от другого узла k и если $u_i + d_{ij} < u_j$, то заменить метку $[u_j, k]$ на $[u_i + d_{ij}, i]$.
  - Если *все* узлы имеют постоянные метки, процесс вычислений заканчивается. В противном случае выбрать метку $[u_r, s]$ с наименьшим значением расстояния среди всех временных меток (если их несколько – выбор произволен). Изменить статус этой метки на постоянную. Положить $i=r$ и повторить шаг i.

Сетевые модели.

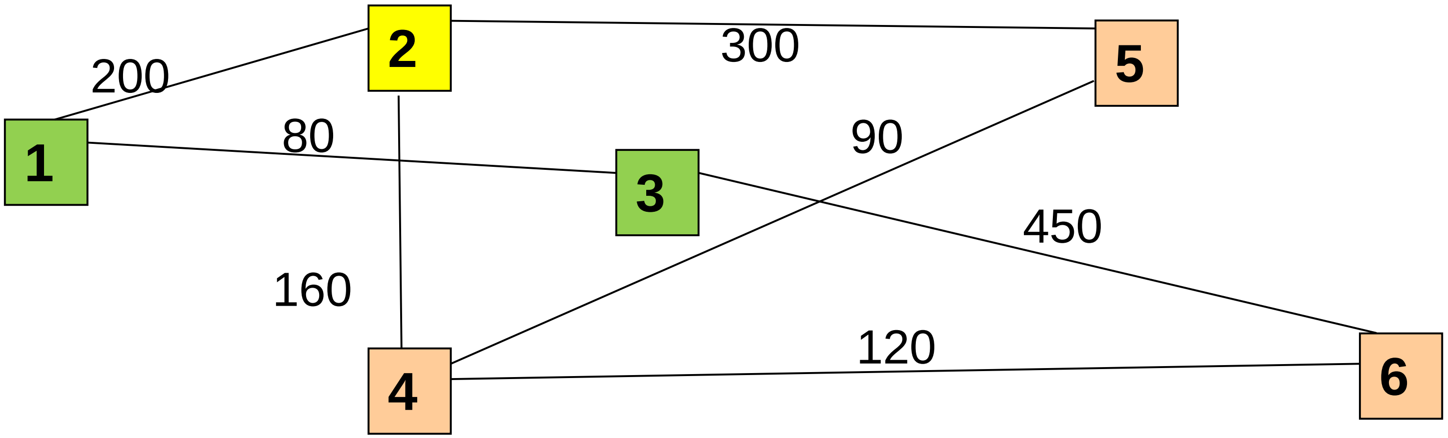
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	временная
3	[0+80,1] = [80,1]	временная
4		
5		
6		

Сетевые модели.

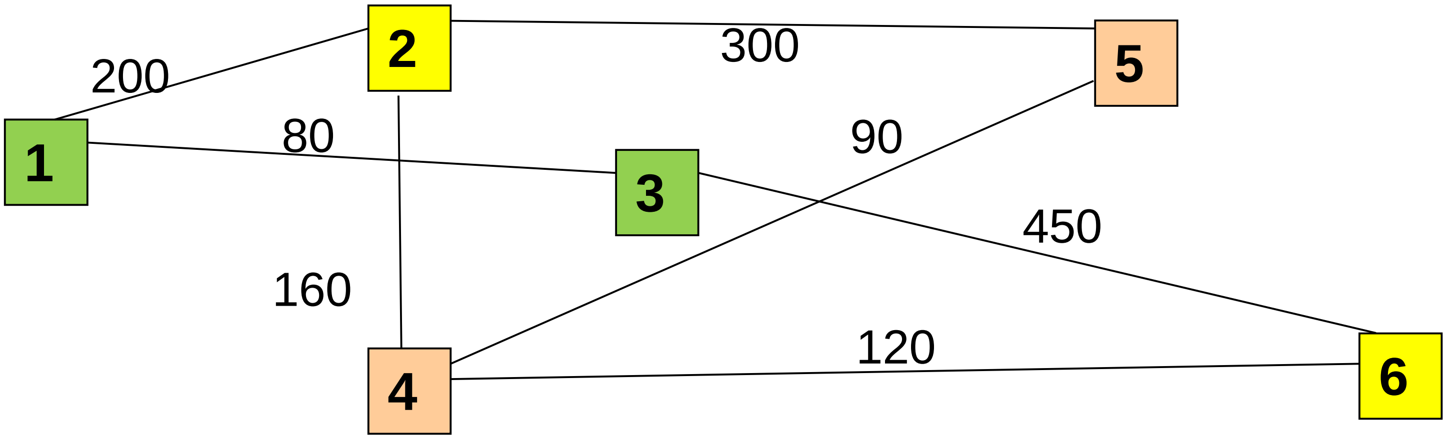
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	временная
3	[0+80,1] = [**80**,1]	постоянная
4		
5		
6		

Сетевые модели.

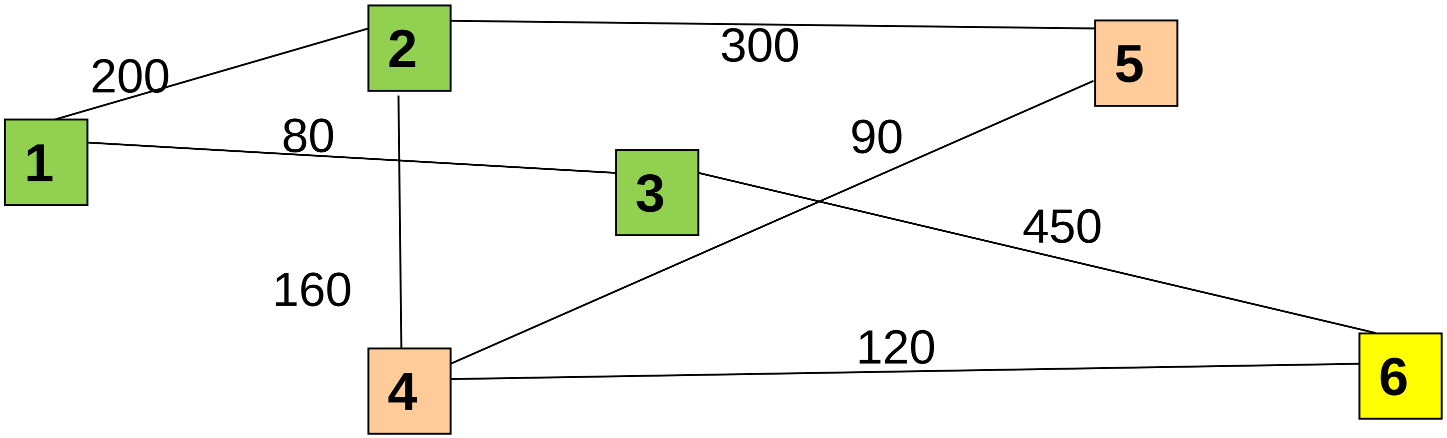
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	временная
3	[0+80,1] = [80,1]	постоянная
4		
5		
6	[80+450] = [530,3]	временная

Сетевые модели.

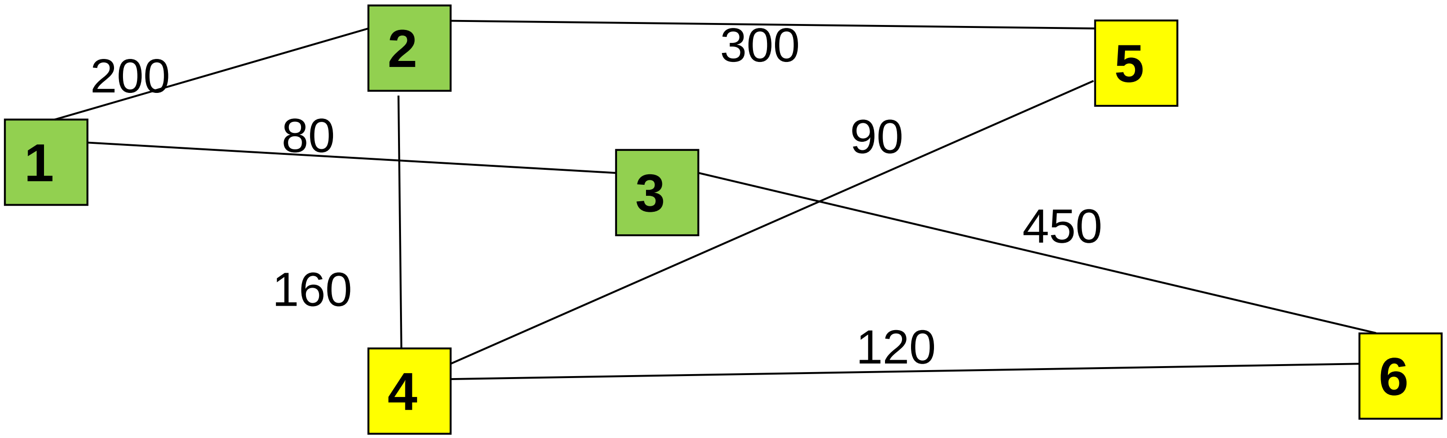
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [**200**,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4		
5		
6	[80+450] = [530,3]	временная

Сетевые модели.

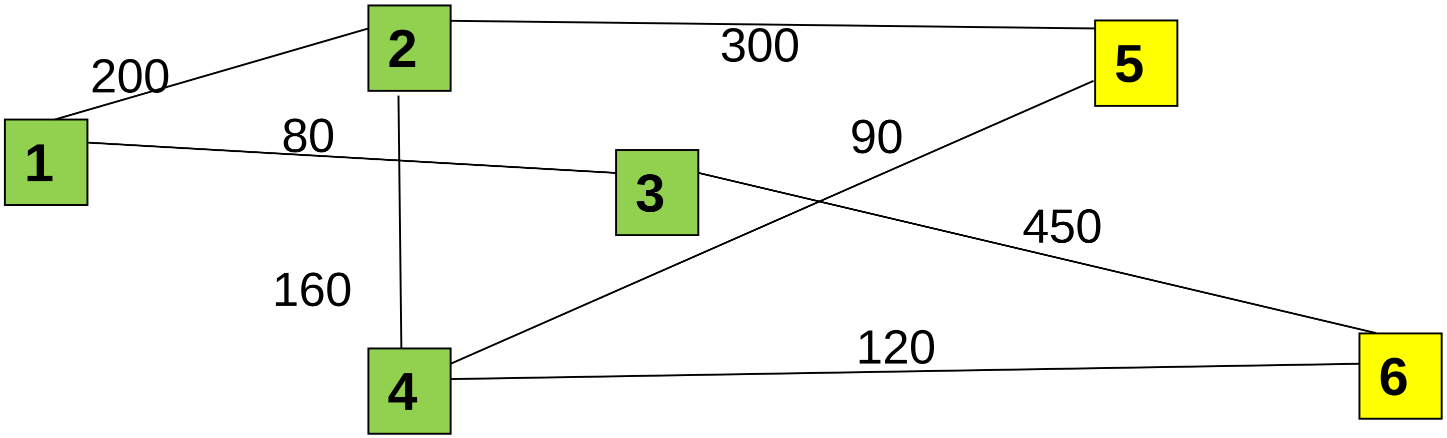
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [360,2]	временная
5	[200+300,2]=[500,2]	временная
6	[80+450] = [530,3]	временная

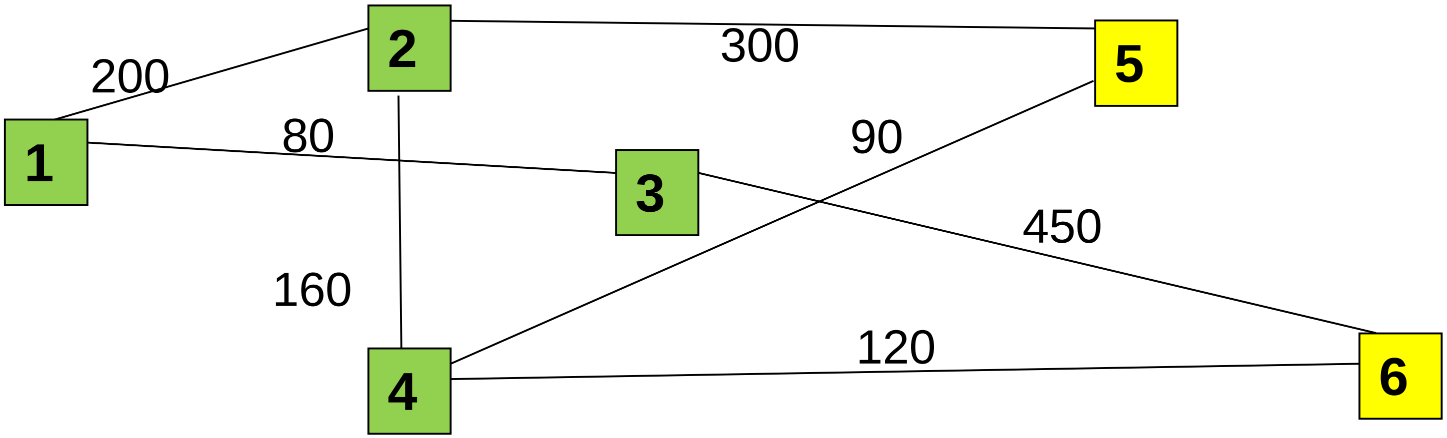
Сетевые модели.

Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [**360**,2]	постоянная
5	[200+300,2]=[500,2]	временная
6	[80+450] = [530,3]	временная

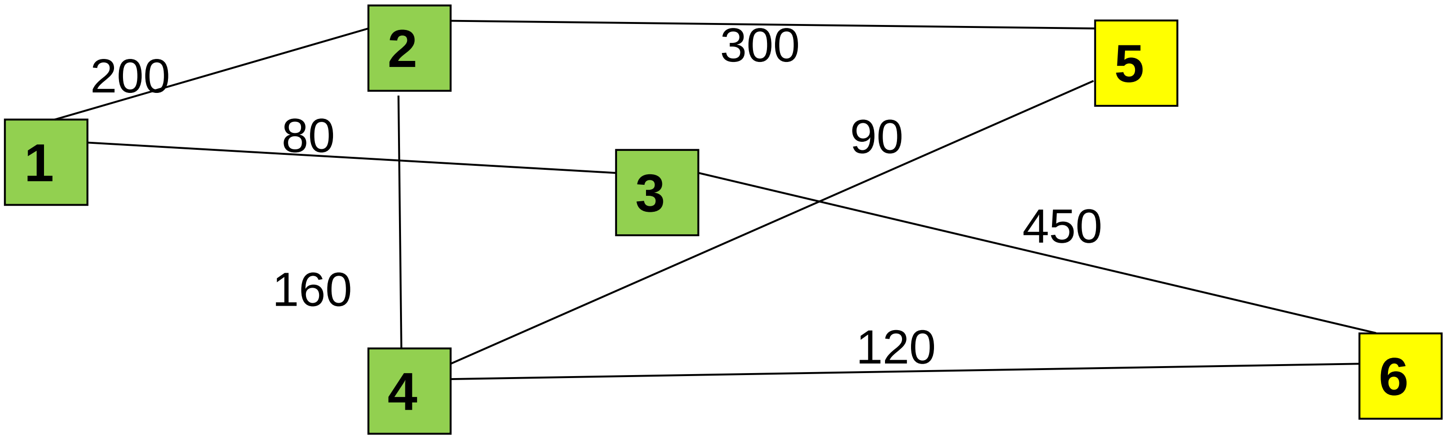
Сетевые модели.
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [360,2]	постоянная
5	360+90<500 => [500,2] <= [360+90,4] = [450,4]	временная
6	360+120<530 =>[530,3]<=[360+120,4] =[480,4]	временная

Сетевые модели.

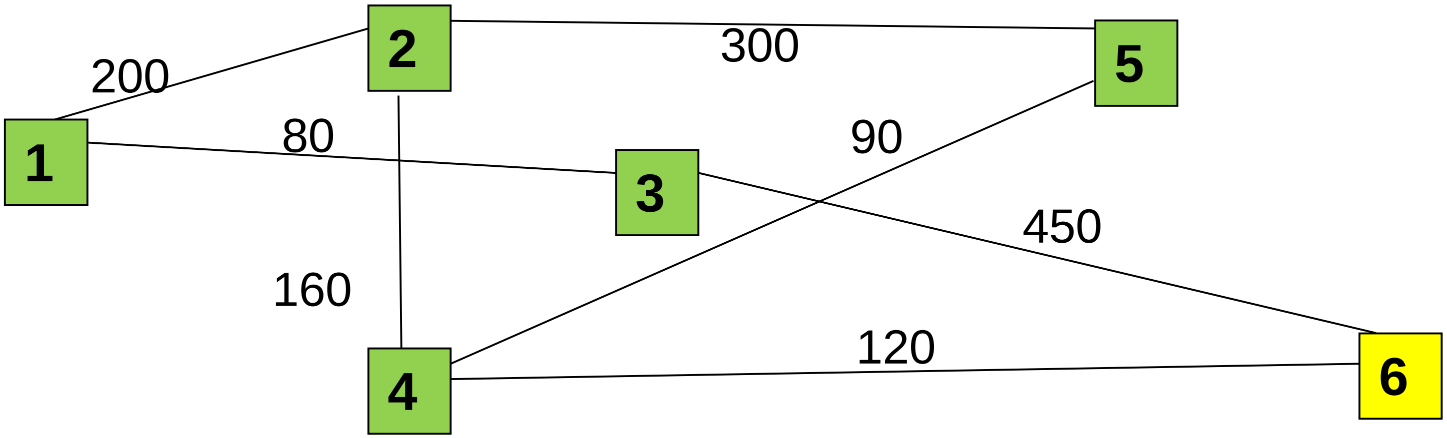
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [360,2]	постоянная
5	[360+90,4] = [450,4]	временная
6	[360+120,4] =[480,4]	временная

Сетевые модели.

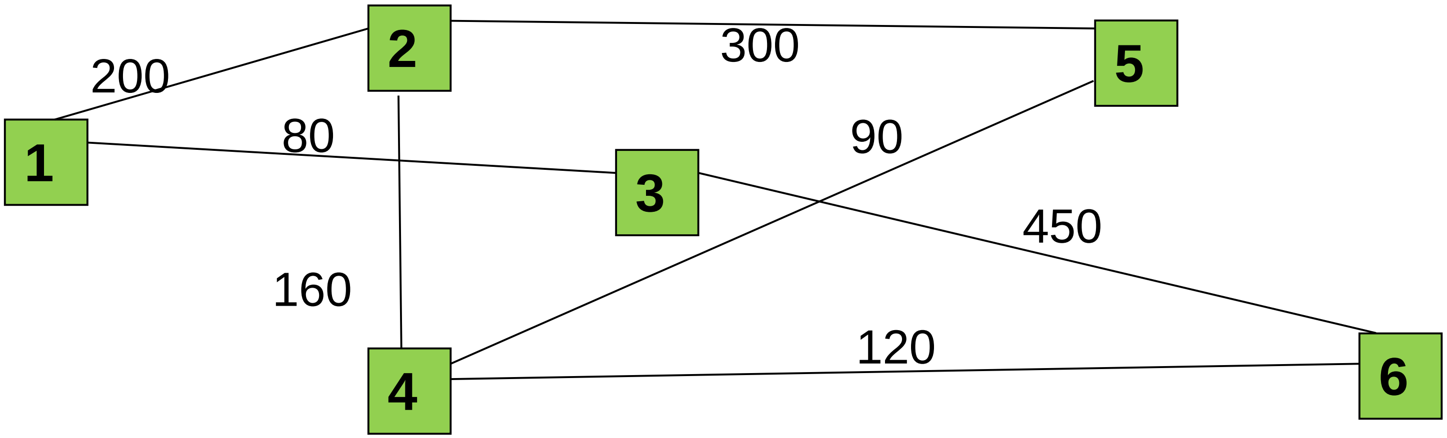
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [360,2]	постоянная
5	[360+90,4] = [**450**,4]	постоянная
6	[360+120,4] =[480,4]	временная

Сетевые модели.

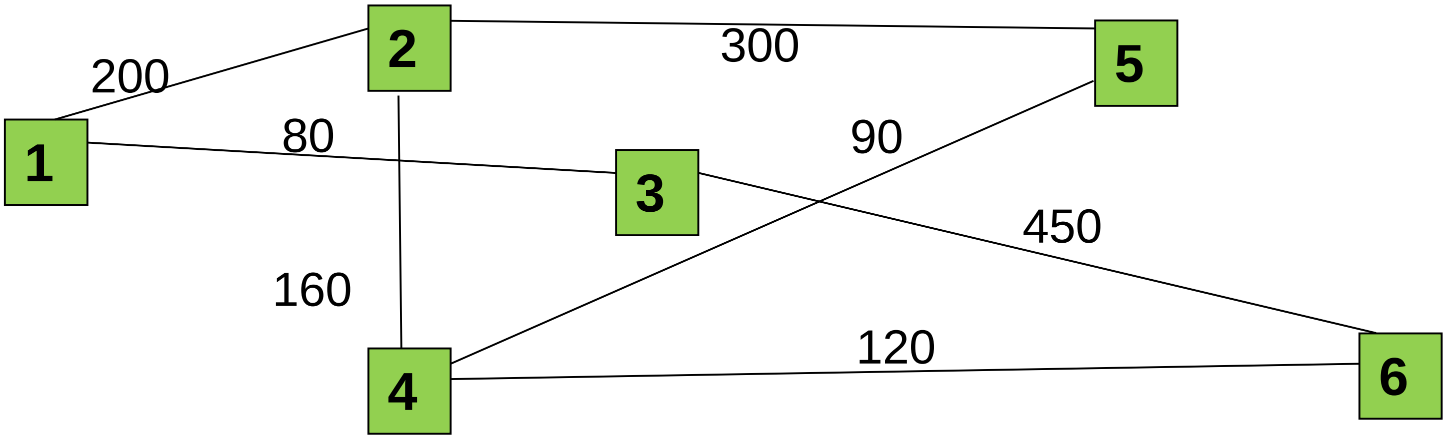
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [360,2]	постоянная
5	[360+90,4] = [450,4]	постоянная
6	[360+120,4] =[480,4]	постоянная

Сетевые модели.

Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [360,2]	постоянная
5	[360+90,4] = [450,4]	постоянная
6	[360+120,4] =[480,4]	постоянная

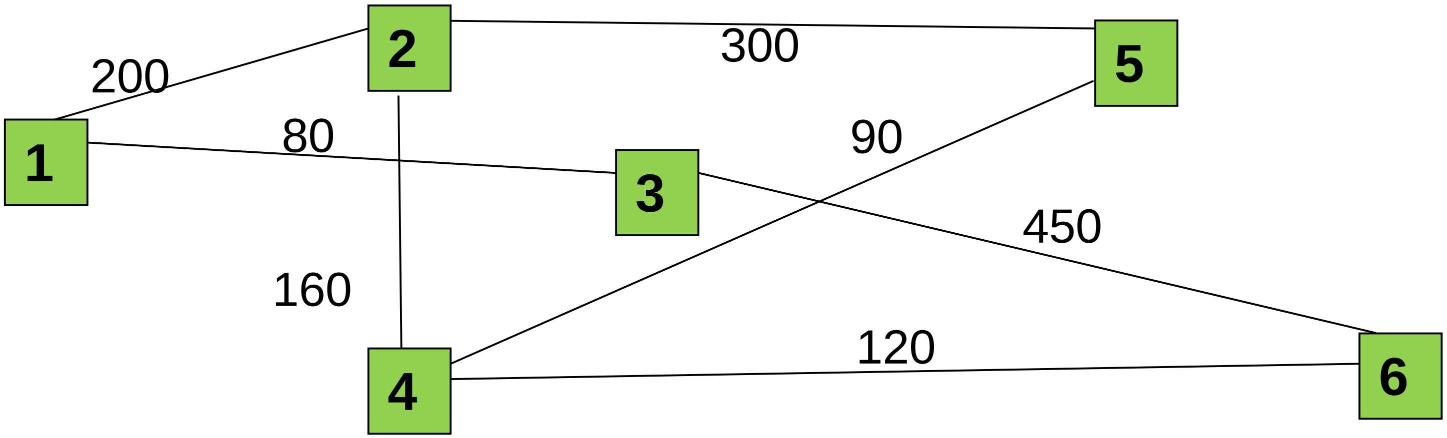
Сетевые модели.
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.

Кратчайший путь между узлом 1 и любым узлом определяется начиная с узла назначения путем прохождения в обратном направлении с помощью информации, представленной в постоянных метках.

Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [360,2]	постоянная
5	[360+90,4] = [450,4]	постоянная
6	[360+120,4] =[480,4]	постоянная

Сетевые модели.

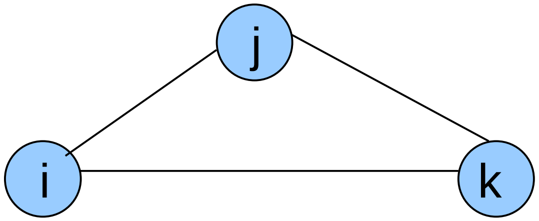
Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+200,1] = [200,1]	постоянная
3	[0+80,1] = [80,1]	постоянная
4	[200+160,2] = [360,2]	постоянная
5	[360+90,4] = [450,4]	постоянная
6	[360+120,4] =[480,4]	постоянная

Сетевые модели.
Нахождение кратчайшего пути. Принцип построения алгоритма Флойда.

- Алгоритм Флойда более общий: он приводит к нахождению кратчайших путей между любыми двумя узлами сети.
- Сеть с n узлами представляется в виде квадратной матрицы с n строками и n столбцами. Элемент (i,j) равен расстоянию d_{ij} от узла i до узла j, которое имеет конечное значение, если узлы связаны дугой и равно бесконечности в противном случае.
- Основная идея метода.
  - Пусть есть три узла i,j,k и заданы расстояния между ними.
  - Если $d_{ij} + d_{jk} < d_{ik}$, то целесообразно заменить путь $i \to k$ путем $i \to j \to k$.
  - Такая замена (ее еще называют треугольный оператор) выполняется систематически в процессе выполнения алгоритма.

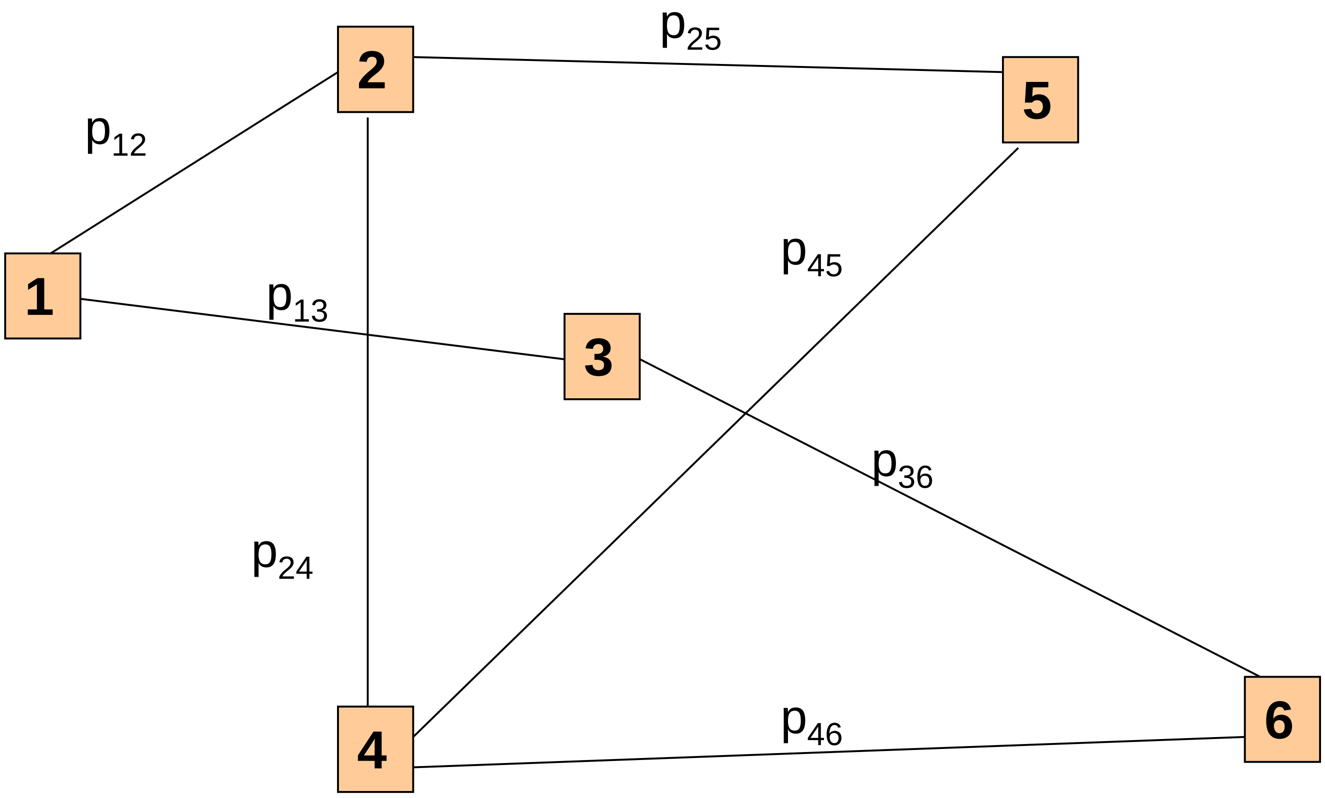


Сетевые модели.
Нахождение кратчайшего пути. Задача минимизации потерь пакетов в корпоративной сети.

- Банк имеет в городе 6 крупных отделений, соединенных оптоволоконными линиями передачи.
- Необходимо организовать видеоконференцсвязь между центральным офисом (узел 1) и отделениями. Для этого необходимо предложить схему статической маршрутизации пакетов, минимизирующую потери пакетов на маршрутах от узла 1 до каждого из остальных узлов. Усредненные значения вероятностей доставки пакетов UDP от узла к узлу в условиях нормальной загрузки сети известны.
- Разумеется, эта задача тоже идеализирована. В частности:
  - не учитывается влияние изменения самого трафика видеоконференций на вероятности доставки пакетов;
  - не учитываются другие параметры, влияющие на качество видеоконференцсвязи.

Сетевые модели.

Нахождение кратчайшего пути. Задача минимизации задержек пакетов в корпоративной сети. Алгоритм Дейкстры.



Сетевые модели.
Нахождение кратчайшего пути. Задача минимизации потерь пакетов в корпоративной сети.

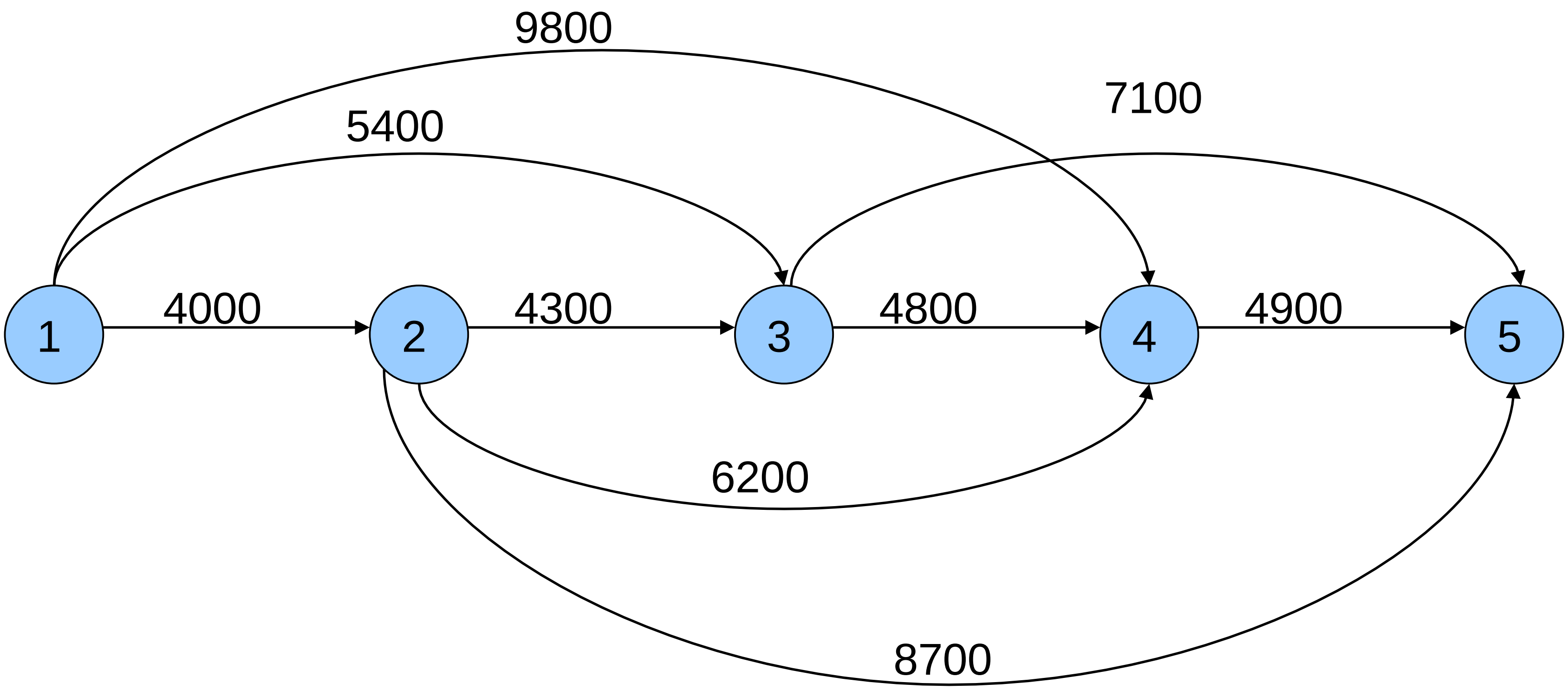
- Мы имеем дело с задачей нахождения не кратчайшего, а наиболее длинного пути.
- Проблема: вероятности не складываются, а умножаются.
- Эта проблема преодолевается, если заменить вероятности их логарифмами: $d_{ij} = \log p_{ij}$.
- Теперь можно воспользоваться алгоритмом Дейкстры или алгоритмом Флойда.

Сетевые модели.
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)

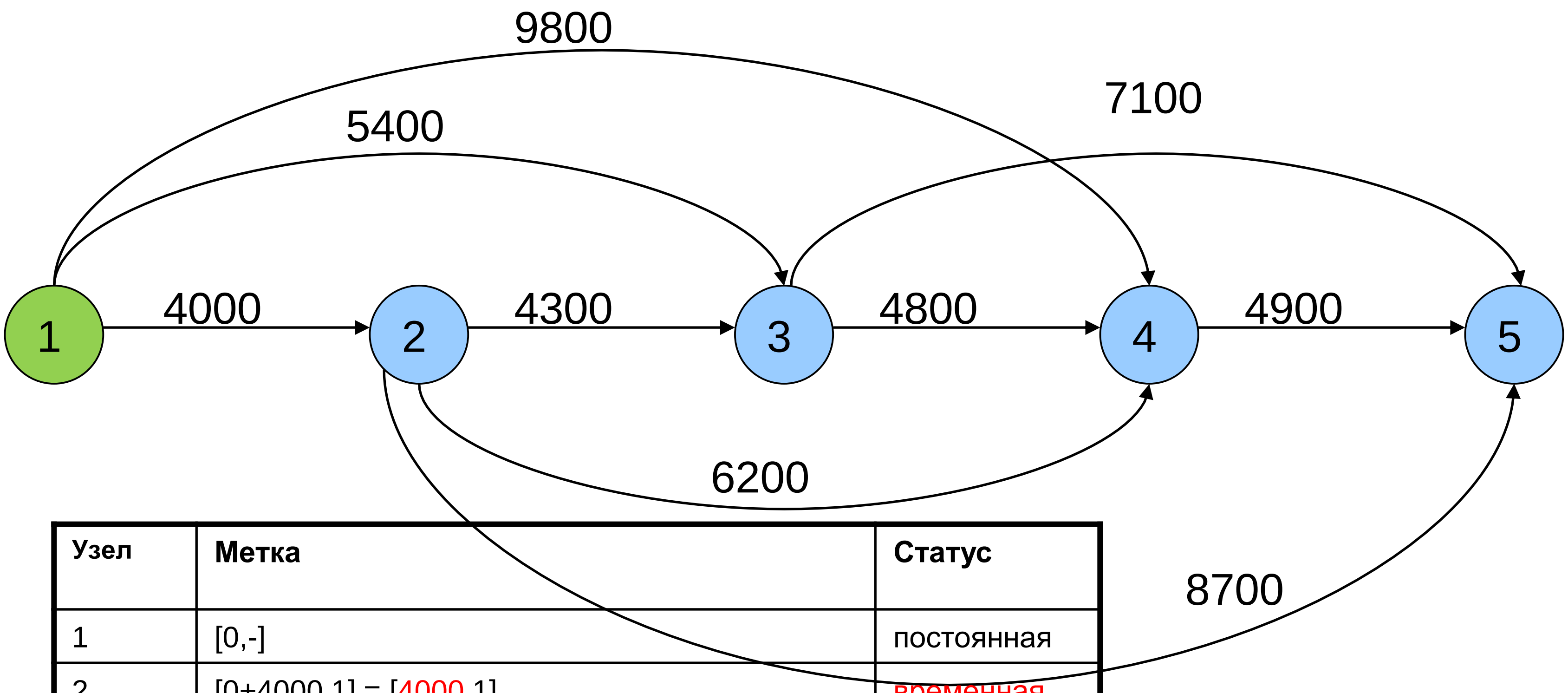
- Компания по прокату автомобилей разрабатывает план обновления парка своих машин на 5 лет (2000-2004 гг.).
- Каждый автомобиль должен прослужить не менее одного и не более трех лет.
- Стоимость замены автомобиля в зависимости от года покупки и срока эксплуатации приведена в таблице.

Год покупки	Стоимость замены в зависимости от срока эксплуатации		
	1 год	2 года	3 года
2000	4000	5400	9800
2001	4300	6200	8700
2002	4800	7100	-
2003	4900	-	-

Сетевые модели.
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)

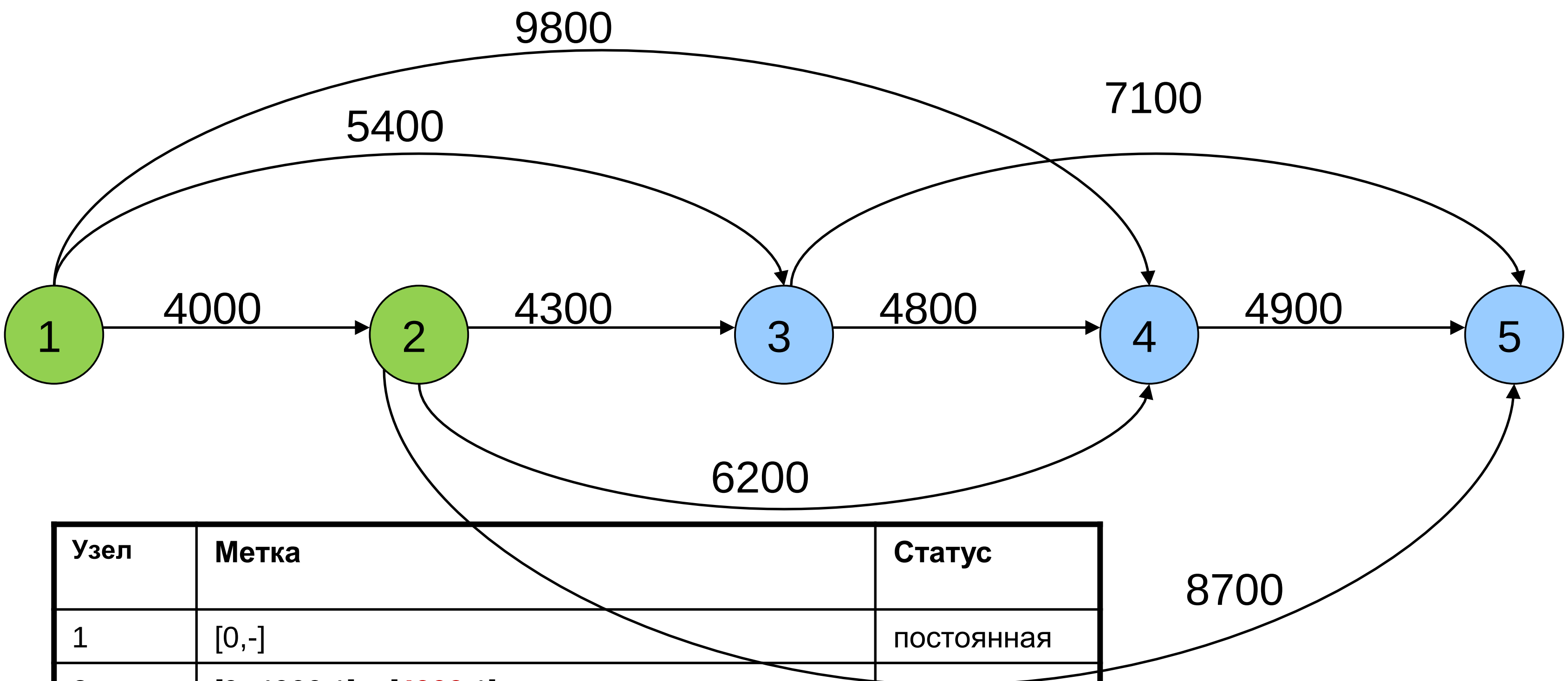


Сетевые модели.
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	временная
3	[0+5400,1] = [5400,1]	временная
4	[0+9800,1] = [9800,1]	временная
5		

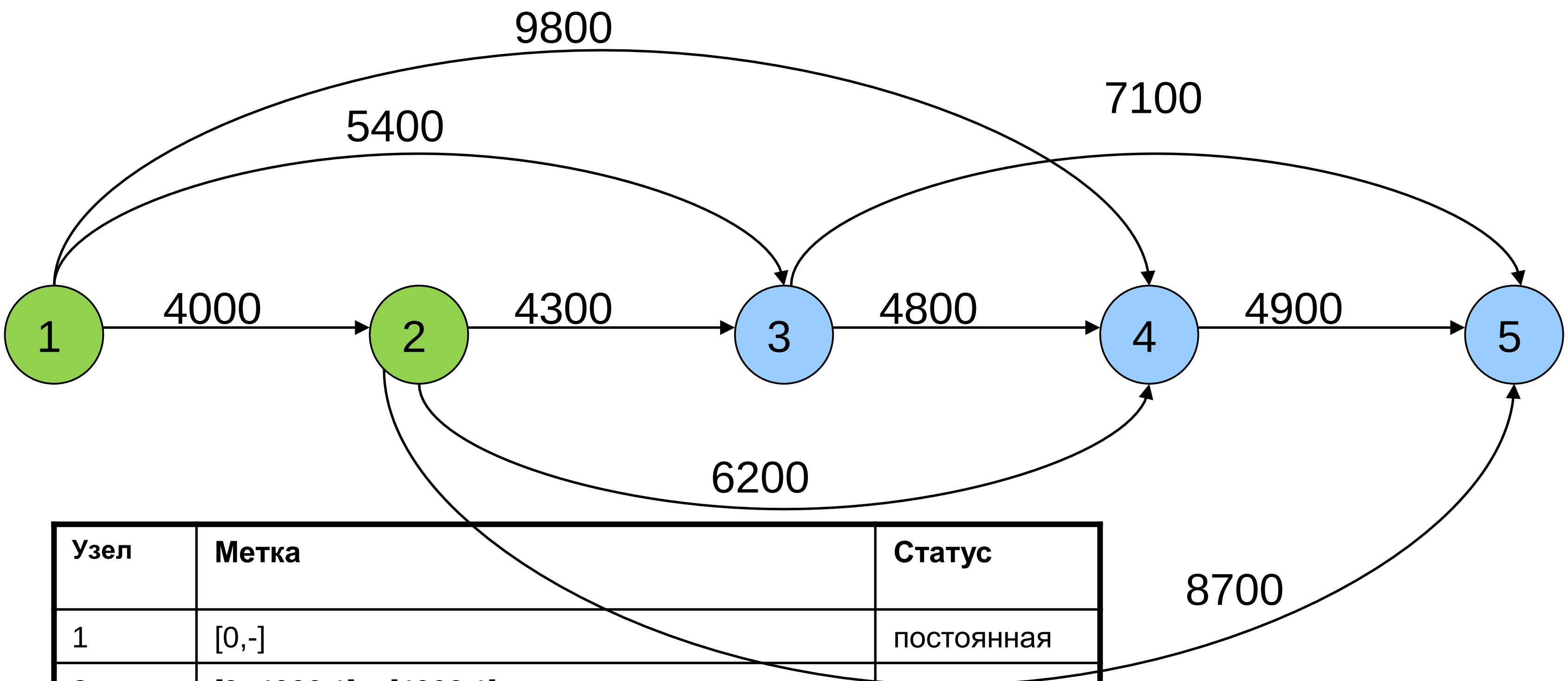
Сетевые модели.
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+4000,1] = [**4000**,1]	постоянная
3	[0+5400,1] = [5400,1]	временная
4	[0+9800,1] = [9800,1]	временная
5		

Сетевые модели.

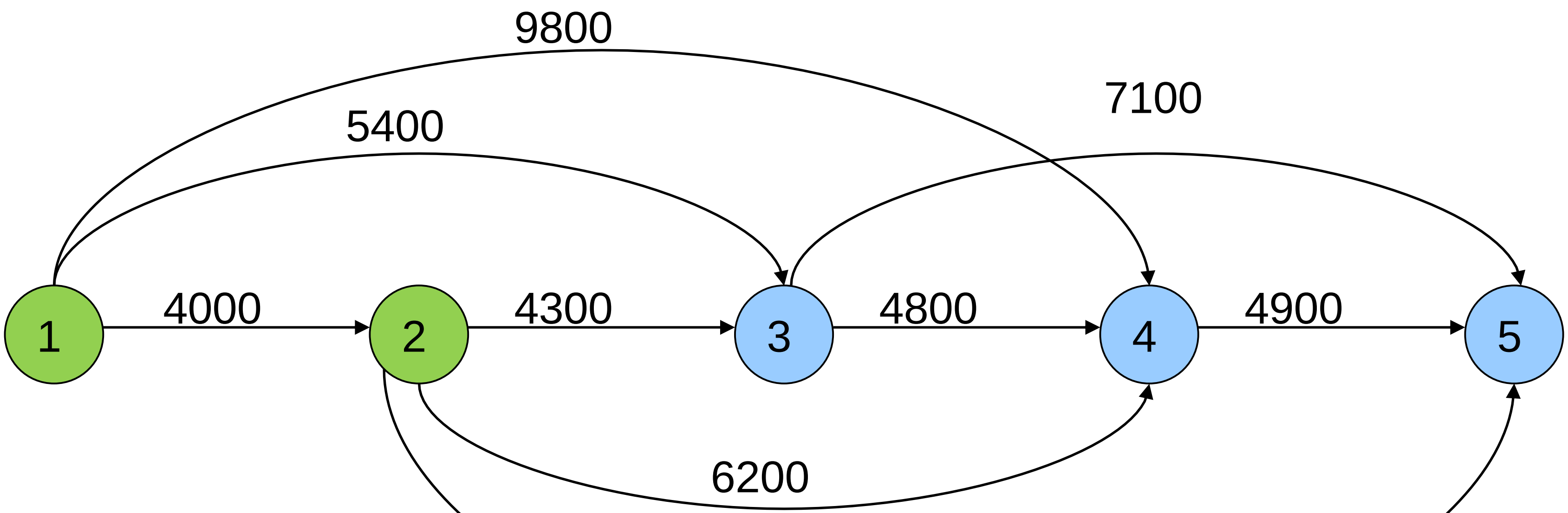
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	4000+4300 > 5400 => [0+5400,1] = [5400,1]	временная
4	4000+6200 > 9800 => [0+9800,1] = [9800,1]	временная
5		

Сетевые модели.

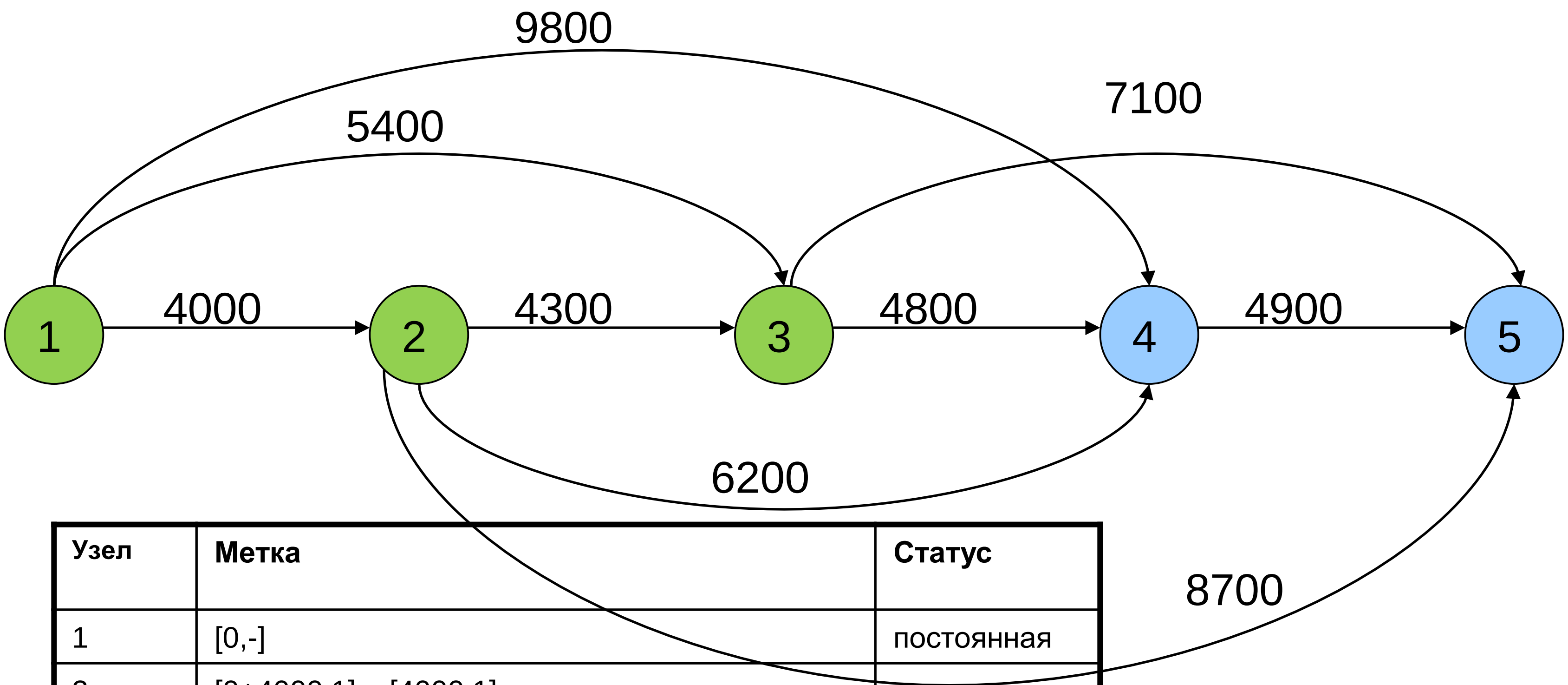
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	Метка	Статус
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	временная
4	[0+9800,1] = [9800,1]	временная
5	[4000+8700,2] = [12700,2]	временная

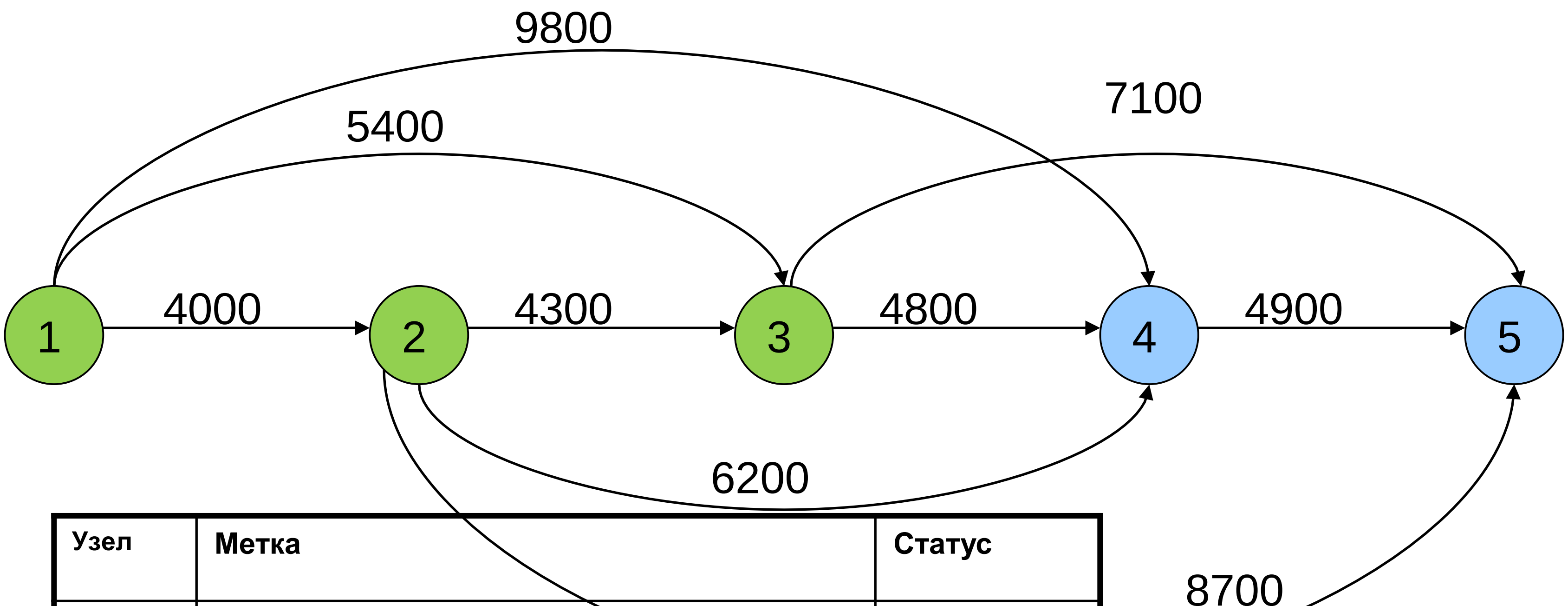
Сетевые модели.

Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [**5400**,1]	постоянная
4	[0+9800,1] = [9800,1]	временная
5	[4000+8700,2] = [12700,2]	временная

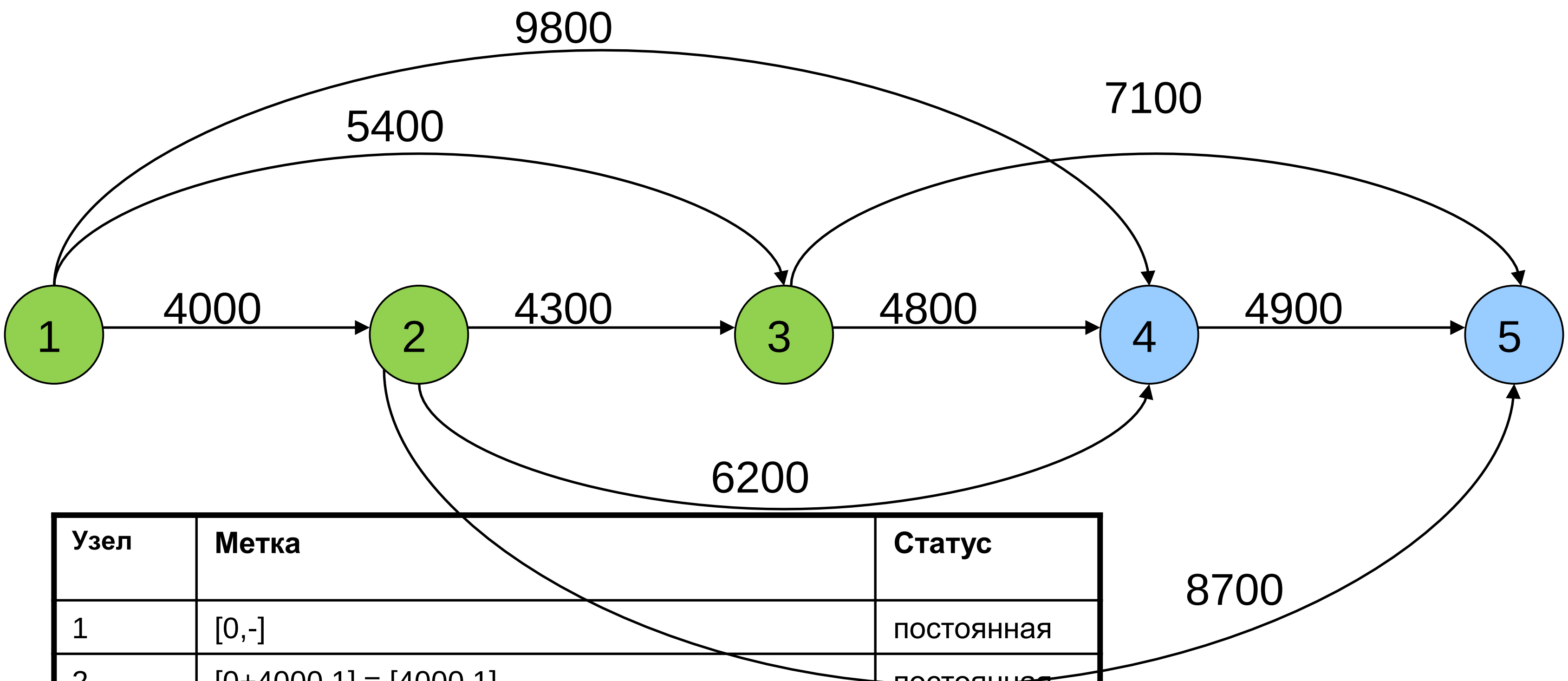
Сетевые модели.
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	Метка	Статус
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	постоянная
4	5400+4800>9800 =>[0+9800,1] = [9800,1]	временная
5	5400+7100 < 12700 => [4000+8700,2] = [12700,2] => [5400+12700, 3] = [12300,3]	временная

Сетевые модели.

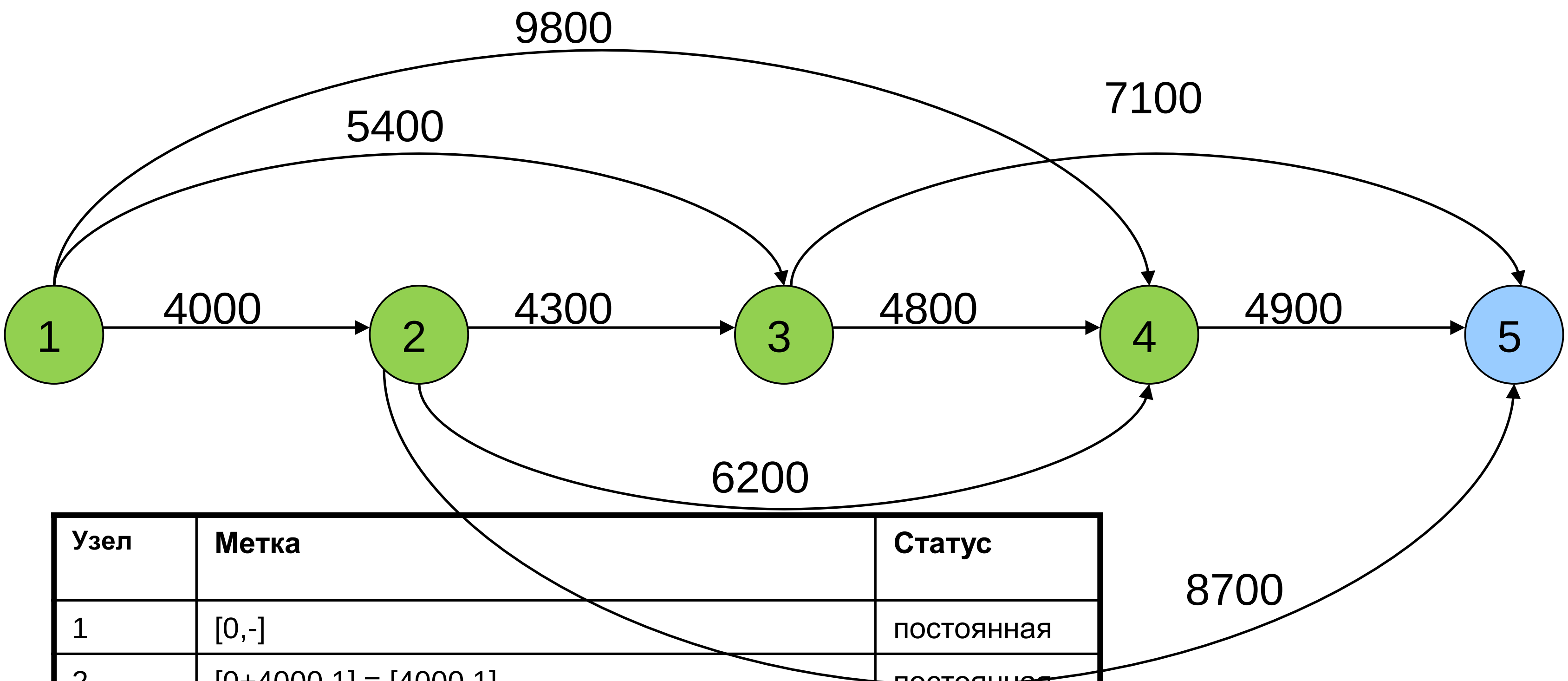
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	Метка	Статус
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	постоянная
4	[0+9800,1] = [9800,1]	временная
5	[5400+12700, 3] = [12300,3]	временная

Сетевые модели.

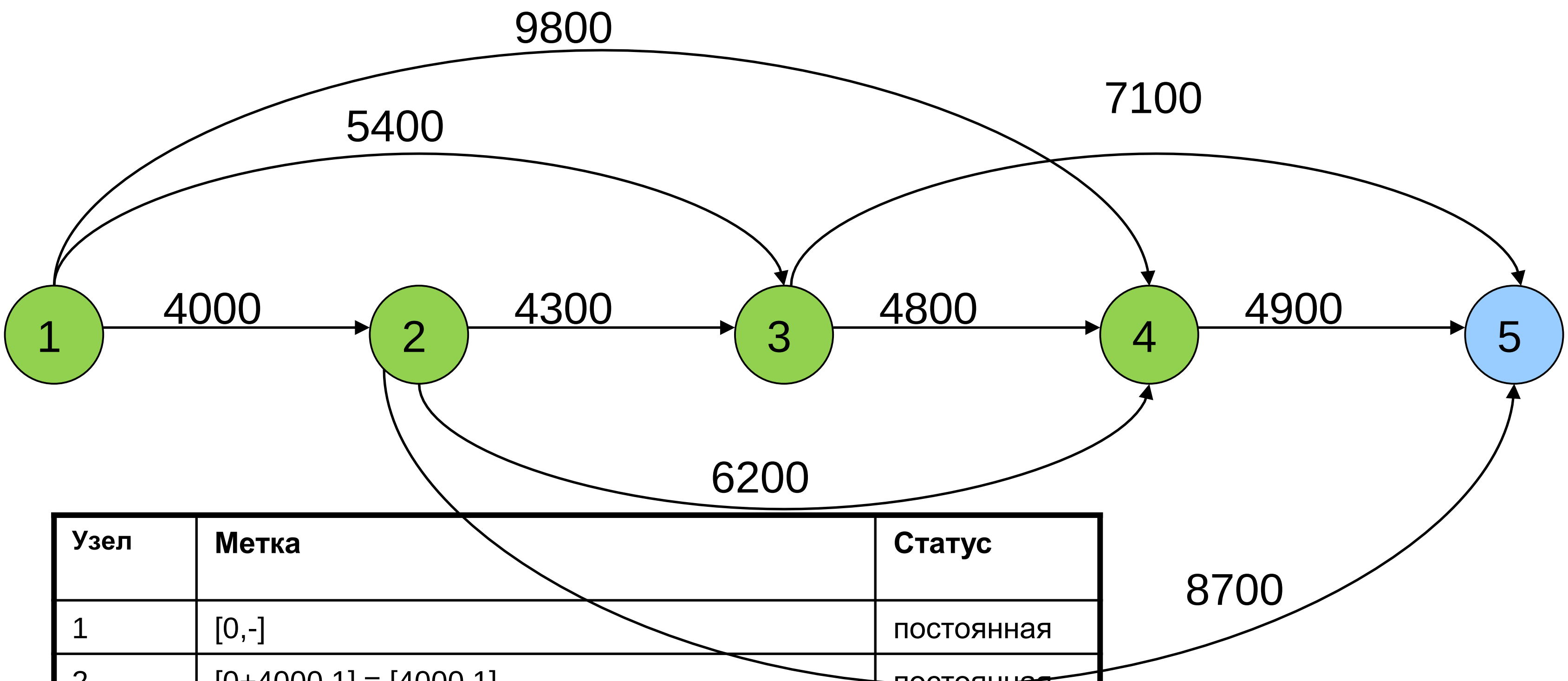
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	Метка	Статус
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	постоянная
4	[0+9800,1] = [**9800**,1]	постоянная
5	[5400+12700, 3] = [12300,3]	временная

Сетевые модели.

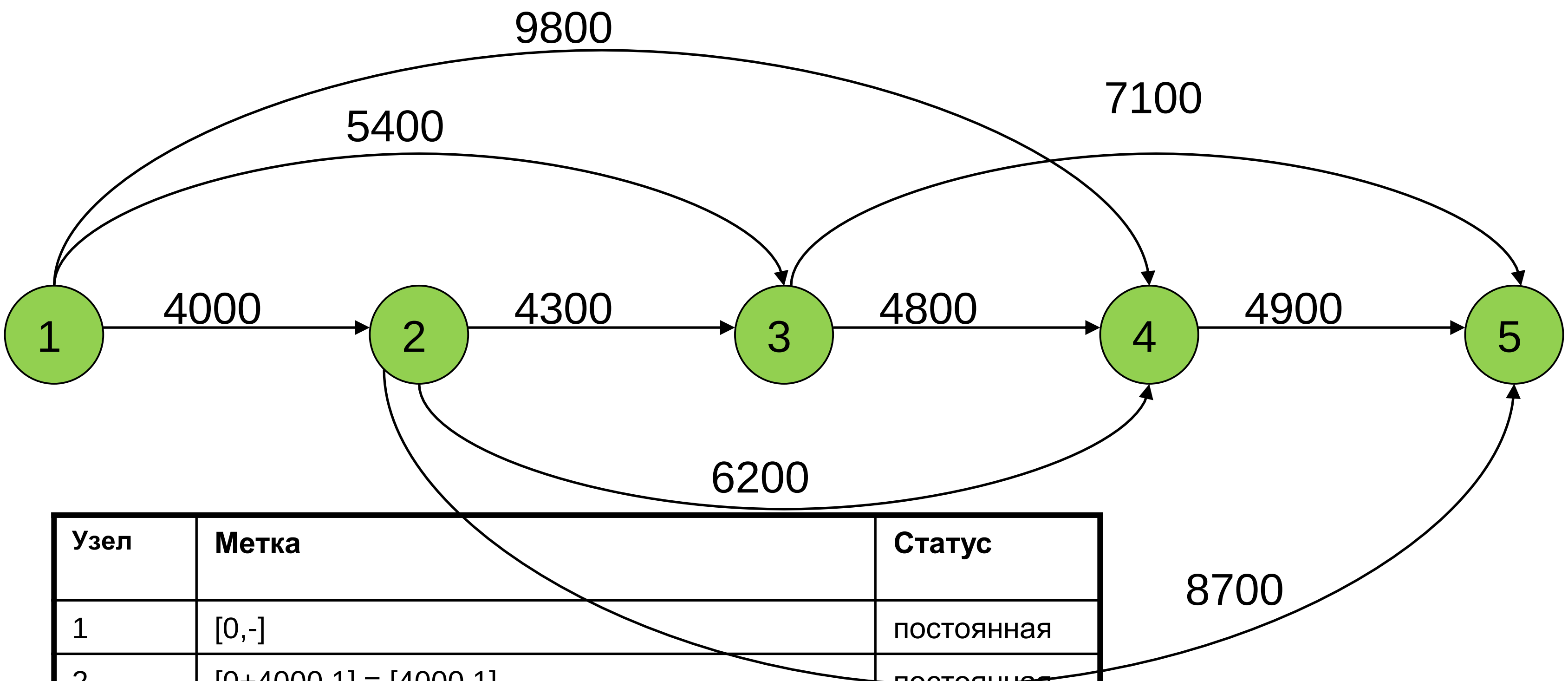
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	Метка	Статус
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	постоянная
4	[0+9800,1] = [9800,1]	постоянная
5	9800+4900 > 12300 => [5400+12700, 3] = [12300,3]	временная

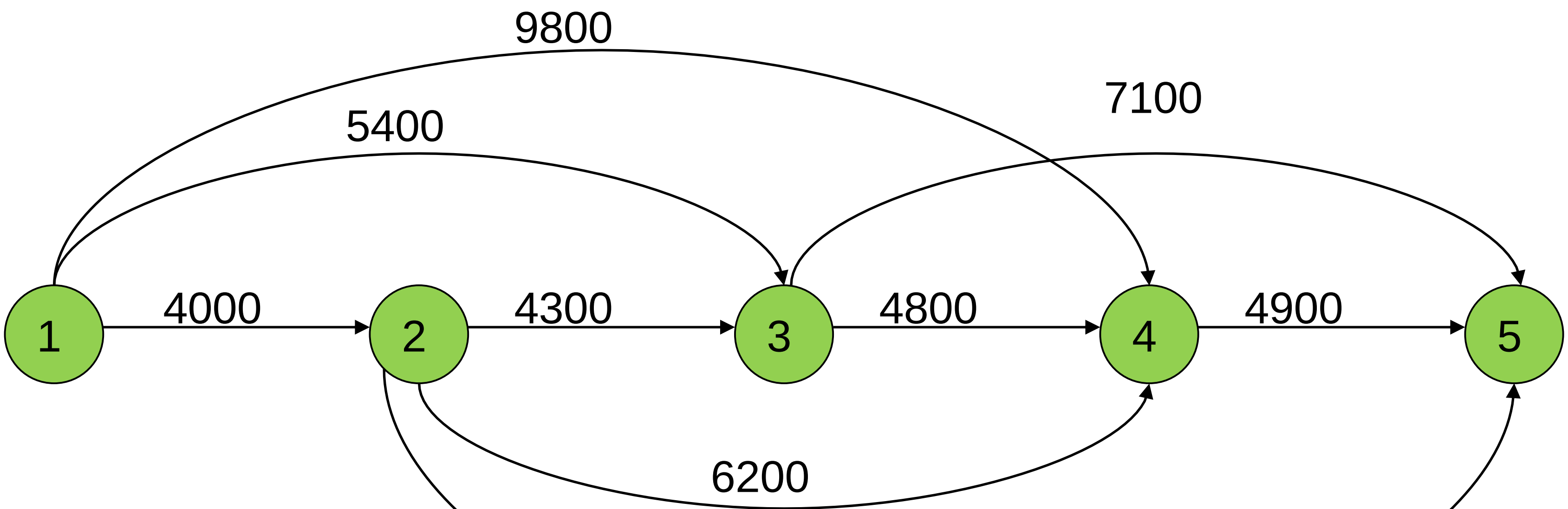
Сетевые модели.

Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	Метка	Статус
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	постоянная
4	[0+9800,1] = [9800,1]	постоянная
5	[5400+12700, 3] = [**12300**,3]	постоянная

Сетевые модели.
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)

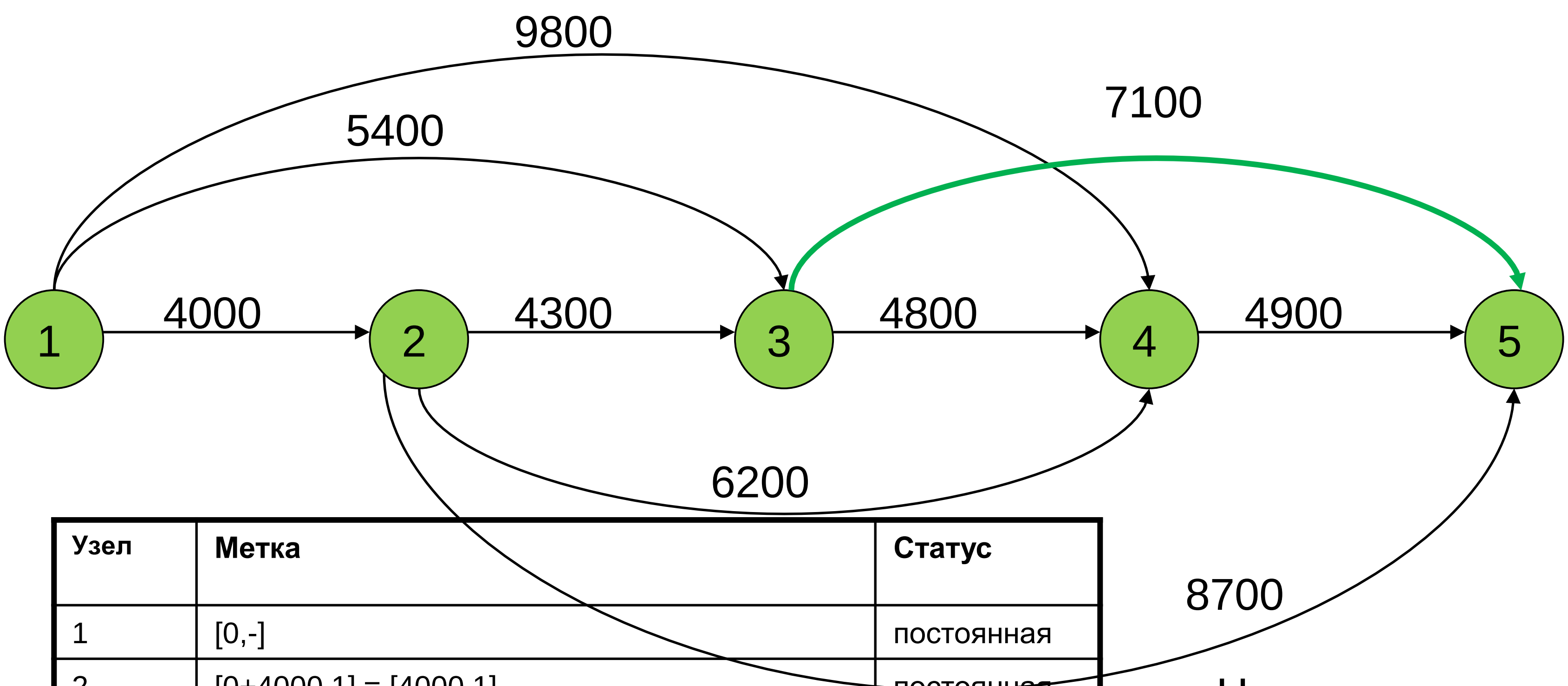


Узел	**Метка**	**Статус**
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	постоянная
4	[0+9800,1] = [9800,1]	постоянная
5	[5400+12700, 3] = [12300,3]	постоянная

Находим кратчайший маршрут из узла 1 в узел 5

Сетевые модели.

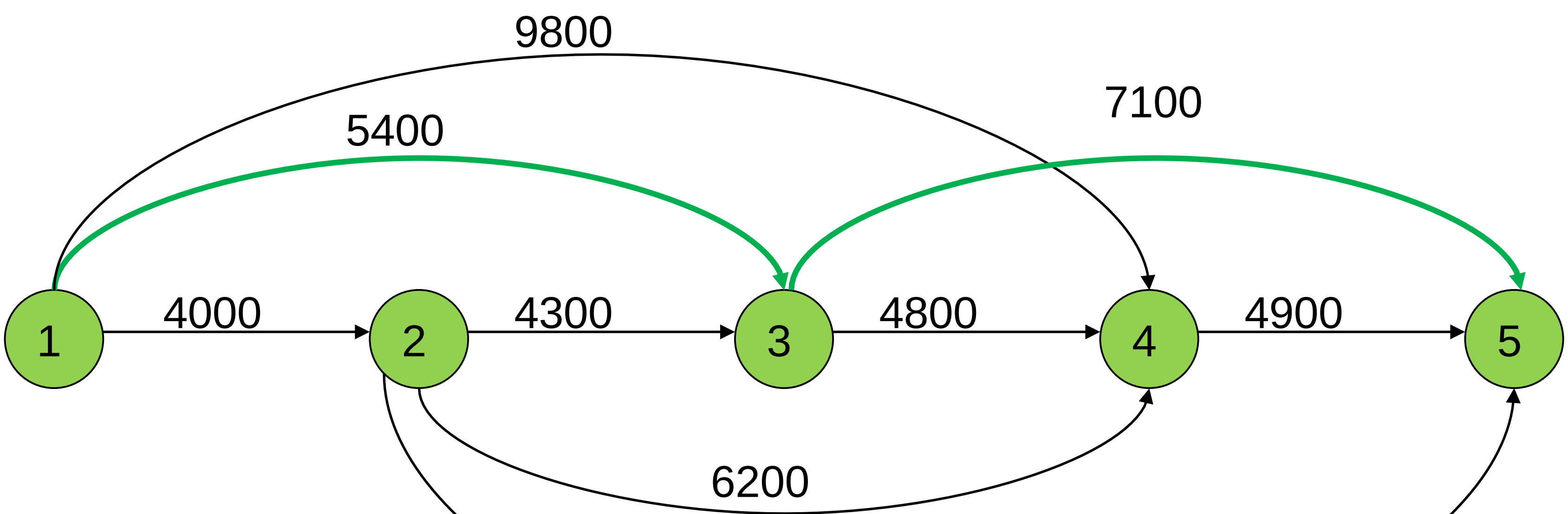
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	Метка	Статус
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	постоянная
4	[0+9800,1] = [9800,1]	постоянная
5	[5400+12700, 3] = [12300,3]	постоянная

Находим кратчайший маршрут из узла 1 в узел 5

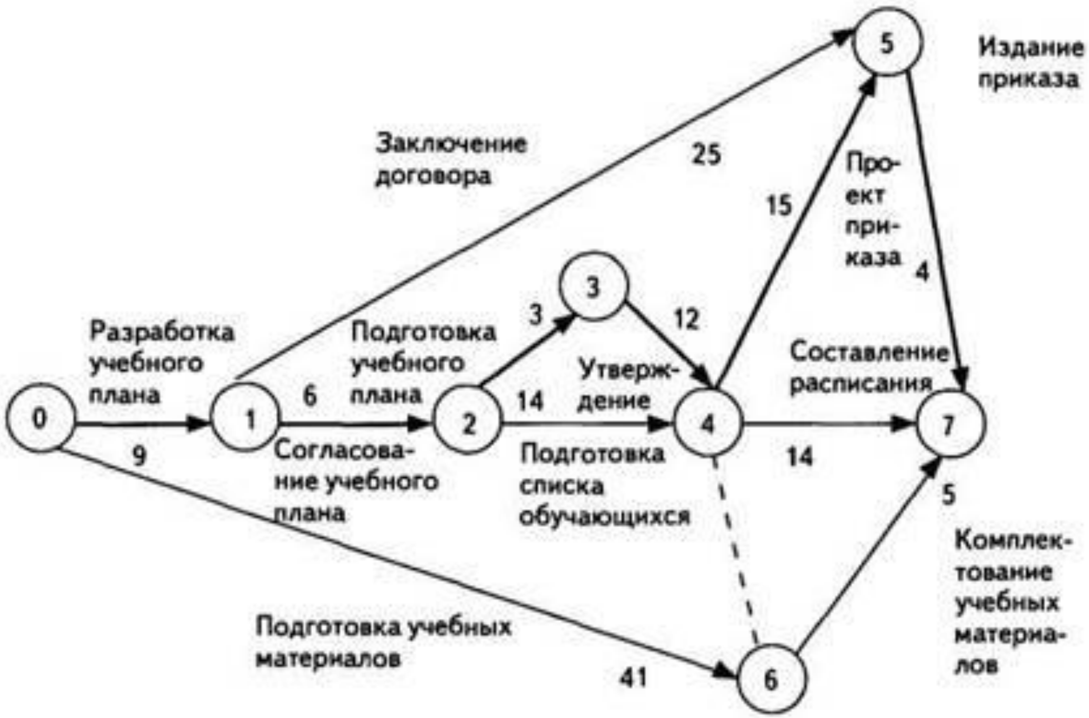
Сетевые модели.
Нахождение кратчайшего пути. Задача о замене оборудования (Х. Таха)



Узел	Метка	Статус
1	[0,-]	постоянная
2	[0+4000,1] = [4000,1]	постоянная
3	[0+5400,1] = [5400,1]	постоянная
4	[0+9800,1] = [9800,1]	постоянная
5	[5400+12700, 3] = [12300,3]	постоянная

Находим кратчайший маршрут из узла 1 в узел 5

Сетевые модели.
Нахождение критического пути. Задача сетевого планирования.



- Критический путь – самый длинный путь на графе.
- Алгоритм нахождения – аналогичен алгоритму Дейкстры.

Сетевые модели.

Нахождение критического пути. Задача сетевого планирования.

$t_k(1)=t_k(0)+t_{01}=0+9=9$

$t_k(2)=t_k(1)+t_{12}=9+6=15$

$t_k(3)=t_k(2)+t_{23}=15+3=18$

$t_k(4)=\max(t_k(2)+t_{24}=15+14=29;\ t_k(3)+t_{34}=18+12=30)=30$

$t_k(5)=\max(t_k(1)+t_{15}=9+25=34;\ t_k(4)+t_{45}=30+15=45)=45$

$t_k(6)=\max(t_k(1)+t_{15}=9+25=34;\ t_k(4)+t_{45}=30+15=45)=45$

$t_k(7)=\max(t_k(4)+t_{47}=30+14=44;\ t_k(5)+t_{57}=45+4=49;\ t_k(6)+t_{67}=45+5=50)=50$

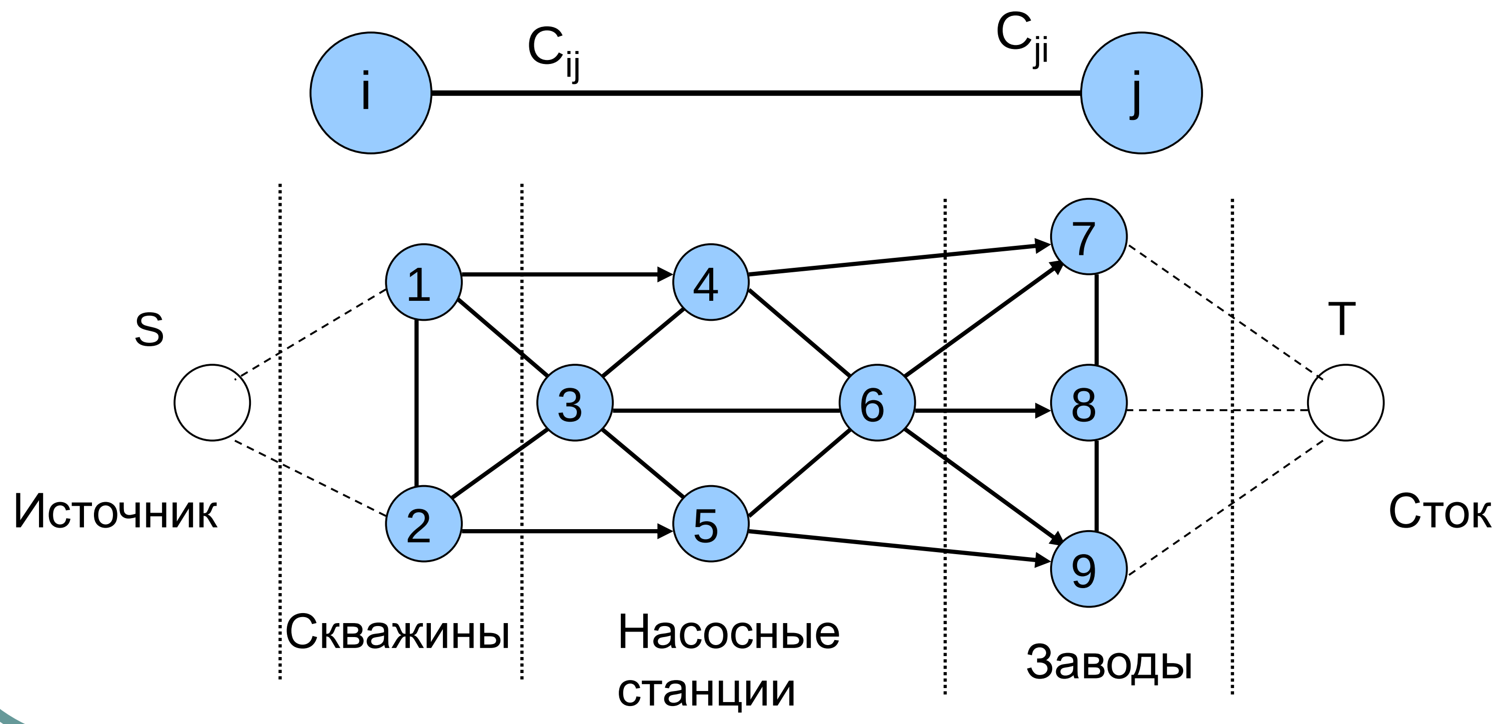
$\mathbf{T_k=50}$

Понятие о постановке и методах решения задач о максимальном потоке и потоке наименьшей стоимости

Сетевые модели.

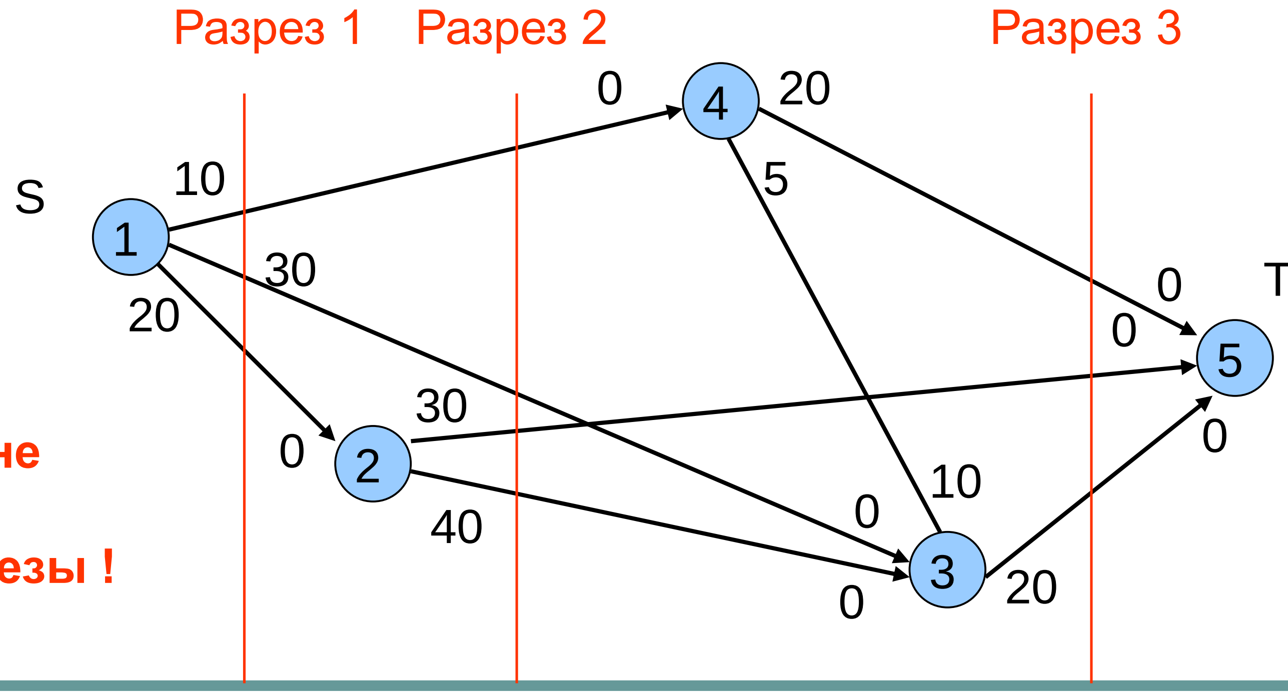
Задача о максимальном потоке. Постановка задачи.

- Рассмотрим сеть трубопроводов для транспортировки сырой нефти от буровых скважин до нефтеперерабатывающих заводов. Каждый сегмент трубопровода имеет свою пропускную способность.
- Сегменты могут быть как однонаправленные, так и двунаправленные.
- Требуется определить максимальный поток между скважинами и заводами.



Сетевые модели.
Задача о максимальном потоке. Перебор разрезов.

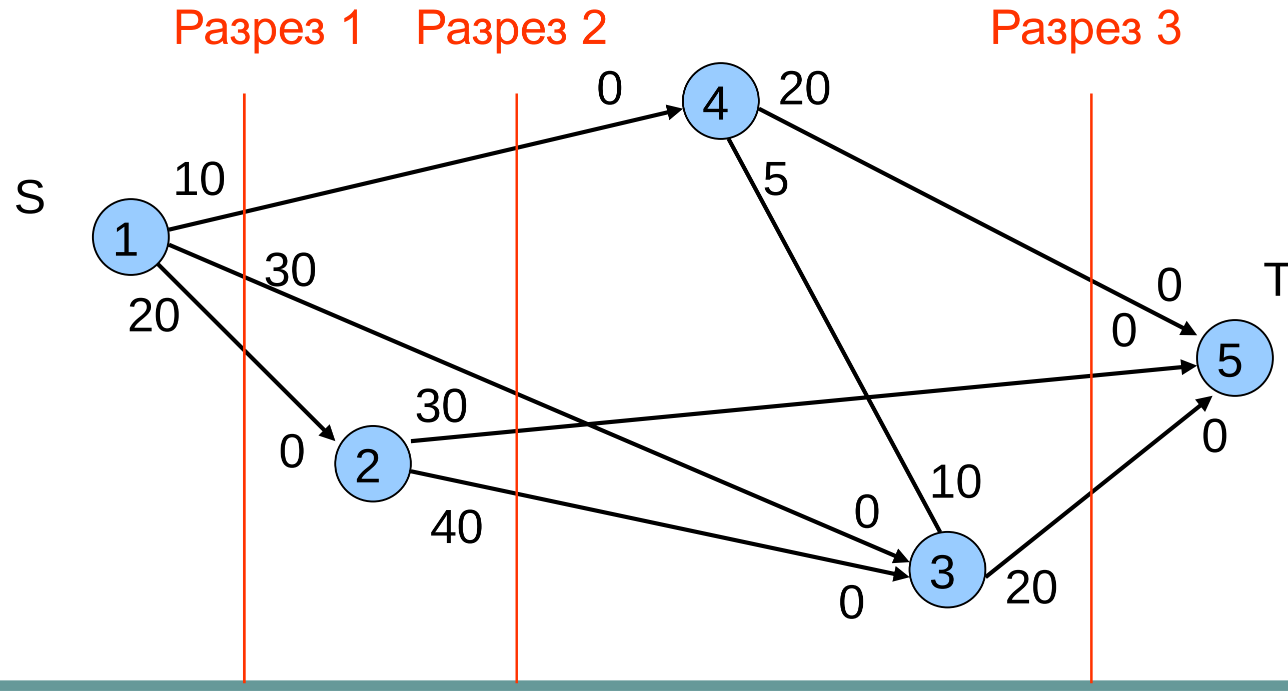
- **Разрез** определяет множество ребер, при удалении которых из сети полностью прекращается поток от источника к стоку.
- **Пропускная способность разреза** равна сумме пропускных способностей разрезанных ребер
- Теорема Форда-Фалкерсона. Для любой сети с источником S и стоком T **максимальная** величина потока из S в T равна **минимальной** пропускной способности разреза.



Это не все разрезы !

Сетевые модели.
Задача о максимальном потоке. Перебор разрезов.

Разрез	Разрезанные ребра	Пропускная способность
1	(1,2), (1,3), (1,4)	10+30+20 = 60
2	(1,3), (1,4), (2,3), (2,5)	30+10+40+30 = 110
3	(2,5), (3,5), (4,5)	30+20+20 = 70



Сетевые модели.
Задача о максимальном потоке. Идея алгоритма.

- **Идея алгоритма** нахождения максимального потока состоит в нахождении сквозных путей с положительными потоками от источника к стоку основана на следствии из теоремы Форда-Фалкерсона: если для системы дуг,задействованных для пропуска потока, нельзя найти ни один новый сквозной путь, увеличивающий поток, то этот поток является максимальным

- Нахождение очередного сквозного пути предполагает задействование части пропусной способообности ребер. Поэтому следующий сквозной путь ищется на остаточной сети.

- Максимальный поток вычисляется как сумма потоков в сквозных путях.

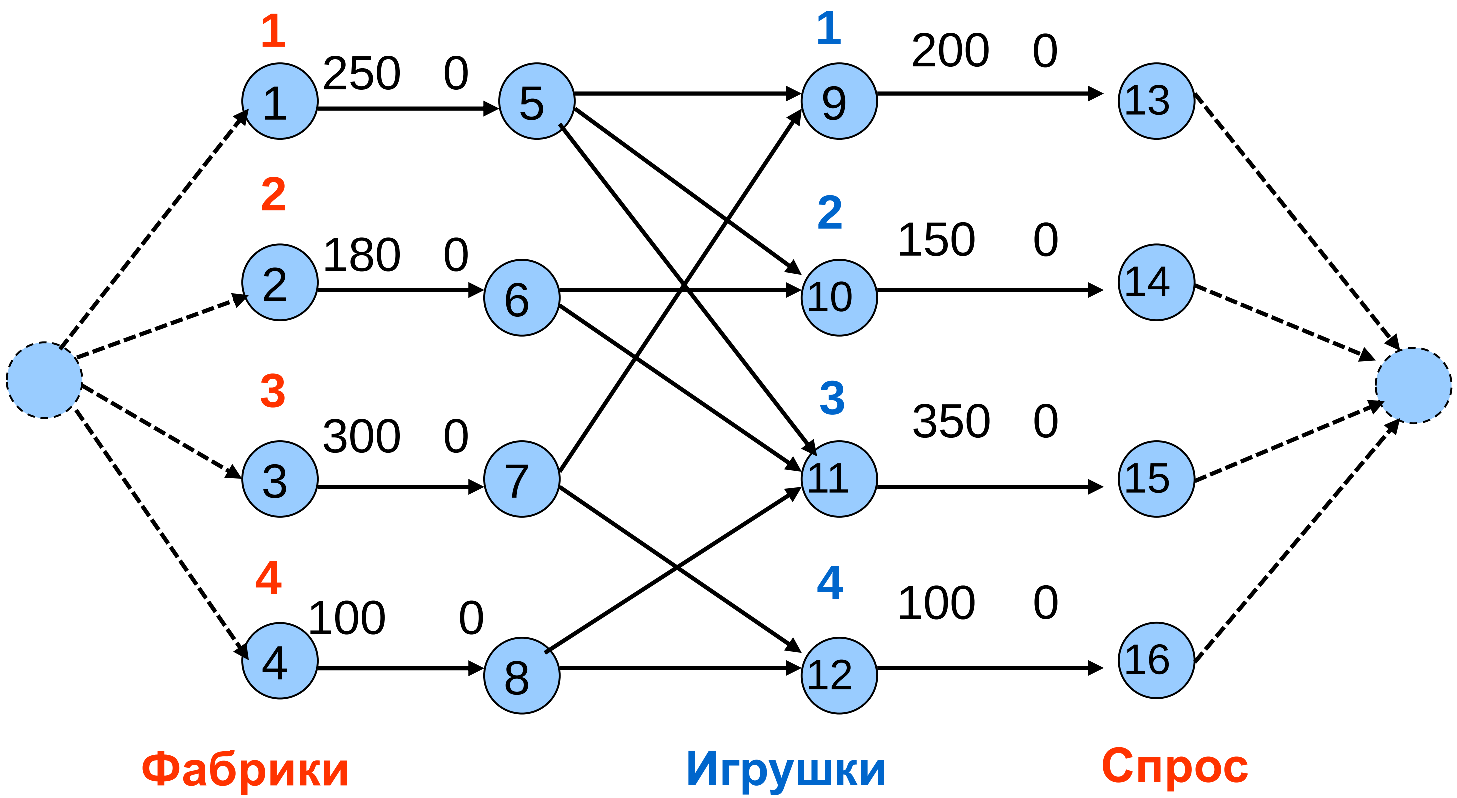
Сетевые модели.
Задача о максимальном потоке.
Оптимизация производственного плана.

- Четыре фабрики имеют заказ на производство четырех видов игрушек.
- Возможности фабрик по производству игрушек показаны в таблице.

Фабрика	Типы игрушек	Ежедневная производственная мощность (игрушек)
1	1,2,3	250
2	2,3	180
3	1,4	300
4	3,4	100

- Ежедневный спрос на игрушки каждого из четырех типов составляет 200, 150, 350 и 100 штук.
- Необходимо разработать производственный план, максимально удовлетворяющий спрос на игрушки

Сетевые модели.
Задача о максимальном потоке.
Оптимизация производственного плана.



- Конечно, задача оптимизации производственного плана может быть сформулирована как классическая задача линейного программирования (причем – целочисленного). Если количество выпускаемых i-й фабрикой игрушек l-го о типа – x_{il}

$$max\ (x_{11}+x_{12}+x_{13}+x_{22}+x_{23}+x_{31}+x_{34}+x_{43}+x_{44});$$

$$x_{11} + x_{12} + x_{13} \le 250; \qquad x_{22} + x_{23} \le 180;$$
$$x_{31} + x_{34} \le 300; \qquad x_{43} + x_{44} \le 100;$$

$$x_{11} + x_{31} \le 200; \qquad x_{12} + x_{22} \le 150;$$
$$x_{13} + x_{23} + x_{43} \le 350; \qquad x_{34} + x_{44} \le 100;$$

Сетевые модели.
Задача о максимальном потоке.
Оптимизация производственного плана.

$$\max\,(x_{11}+x_{12}+x_{13}+x_{22}+x_{23}+x_{31}+x_{34}+x_{43}+x_{44});$$

$$x_{11}+x_{12}+x_{13}\le 250; \qquad x_{22}+x_{23}\le 180;$$
$$x_{31}+x_{34}\le 300; \qquad x_{43}+x_{44}\le 100;$$

$$x_{11}+x_{31}\le 200; \qquad x_{12}+x_{22}\le 150;$$
$$x_{13}+x_{23}+x_{43}\le 350; \qquad x_{34}+x_{44}\le 100;$$

- количество выпускаемых i-й фабрикой игрушек l-го о типа – x_{il}

$x_{11}=0;$ $x_{12}=0;$ $x_{13}=250;$

$x_{22}=150;$ $x_{23}=30$

$x_{31}=200;$ $x_{34}=100;$

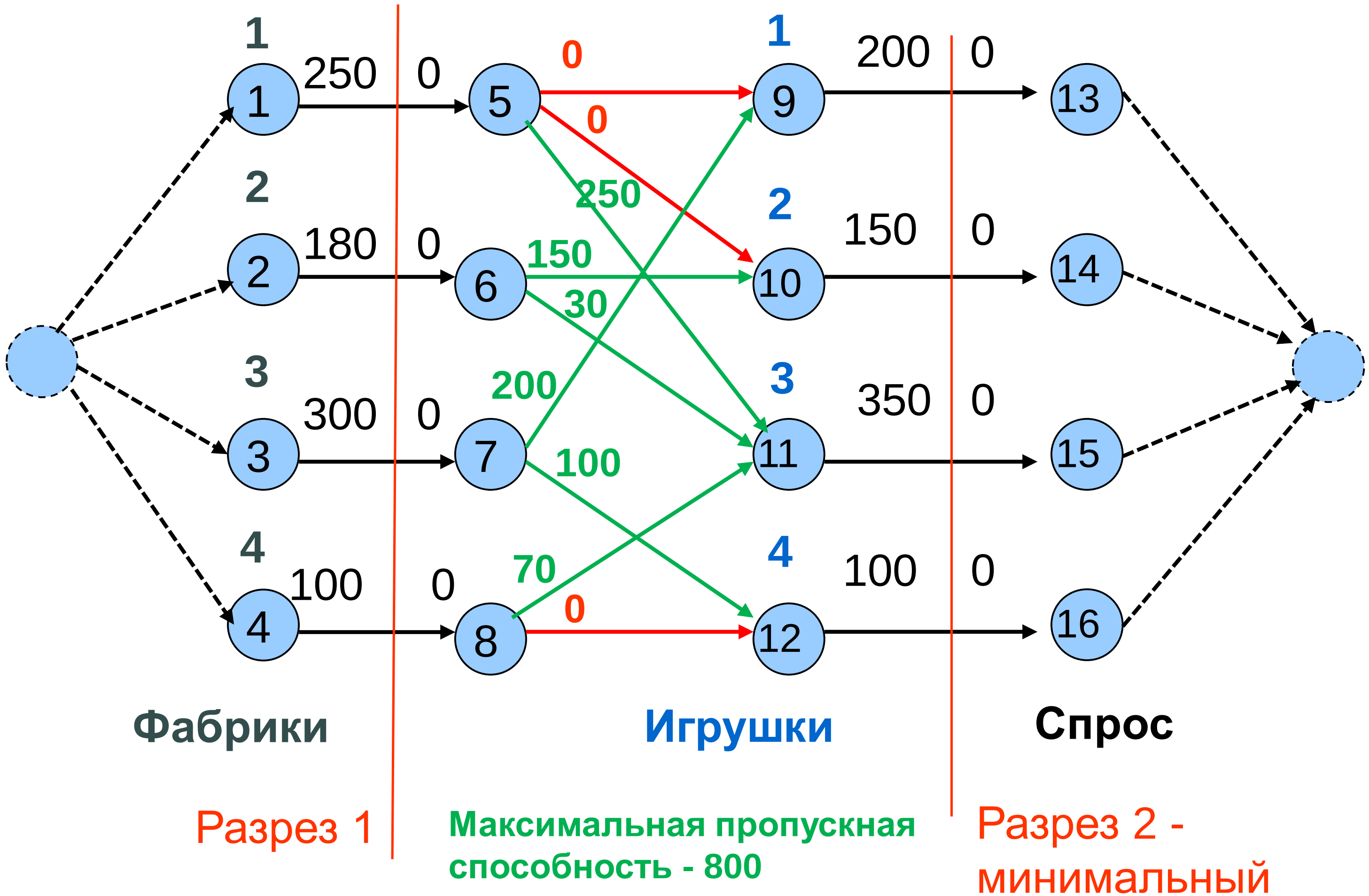
$x_{43}=70;$ $x_{44}=0.$

$F=800$

Сетевые модели.

Задача о максимальном потоке.

Оптимизация производственного плана.



Сетевые модели.
Нахождение потока наименьшей стоимости.
Постановка задачи.

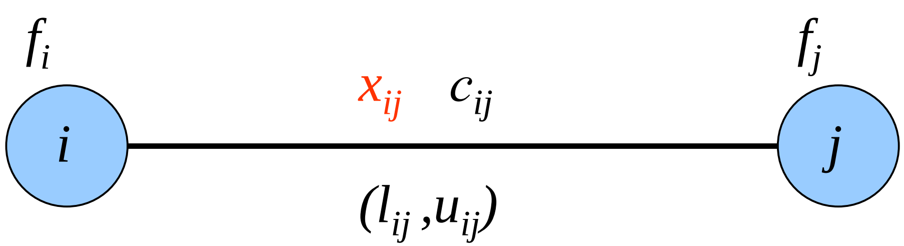
- Задачу нахождения потока наименьшей стоимости в сети с ограниченной пропускной способностью можно рассматривать как обобщение задачи определения максимального потока:
  - Все ребра допускают только одностороннее направление потока, т.е. являются (ориентированными) дугами.
  - Каждой дуге поставлена в соответствие (неотрицательная) стоимость прохождения единицы потока по данной дуге.
  - Дуги могут иметь положительную нижнюю границу пропускной способности.
  - Любой узел сети может выступать в качестве источника и стока.
- В рассматриваемой задаче необходимо найти потоки по дугам, минимизирующие стоимость прохождения потока по сети. При этом должны удовлетворяться ограничения на пропускные способности дуг и на величины предложений и спроса отдельных (или всех) узлов.

Сетевые модели.

Нахождение потока наименьшей стоимости.

Постановка задачи.

- Рассматривается сеть $G=(N,A)$ с ограниченной пропускной способностью, где N – множество узлов, A – множество дуг. Обозначим:
  - x_{ij} – величина потока, протекающего от узла i к узлу j,
  - u_{ij} – верхняя пропускная способность дуги (i,j),
  - l_{ij} – нижняя пропускная способность дуги (i,j),
  - c_{ij} – стоимость прохождения потока по дуге (i,j),
  - f_i – величина результирующего потока, потребляемого узлом i или производимого узлом i.



Сетевые модели.

Нахождение потока наименьшей стоимости.

Сетевая модель как задача линейного программирования

- Используя данные выше определения, можно записать задачу ЛП для сети с ограниченной пропускной способностью следующим образом:

$$\min \sum_{(i,j\in N)} \sum c_{ij} x_{ij};$$

$$\sum_{\substack{k \\ (k\in N)}} x_{jk} - \sum_{\substack{i \\ (i\in N)}} x_{ij} = f_j, \quad j \in N;$$

$$l_{ij} \le x_{ij} \le u_{ij}.$$

- Условие сбалансированности сети: $\sum f_i = 0$. Сбалансированность сети не гарантирует существования допустимого решения: этому может помешать ограниченность пропускных способностей дуг.

Сетевые модели.

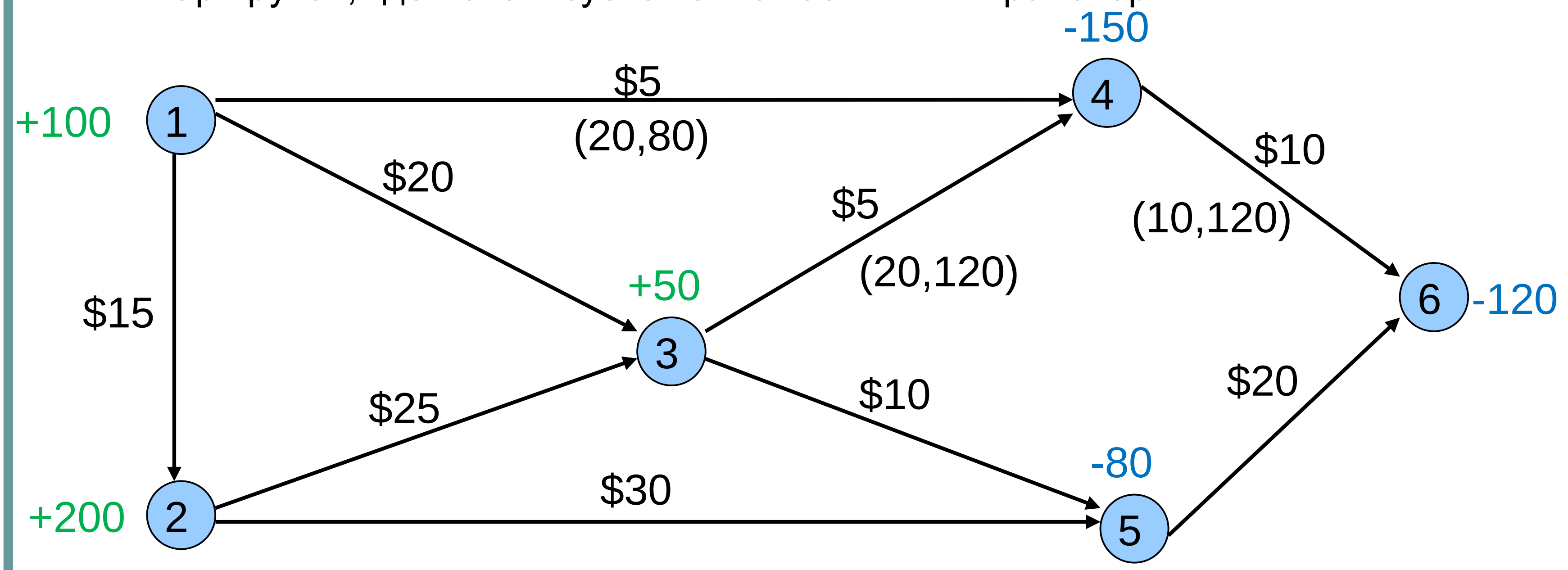
Нахождение потока наименьшей стоимости.

Сетевая модель как задача линейного программирования

- Алгоритм решения базируется на симплекс-методе.
- Модификация алгоритма симплекс-метода заключается в особых правилах ввода и исключения переменных (дуг), то есть в особых условиях оптимальности и допустимости, облегчающих процесс вычислений.
- Базисному решению соответствует минимальное остовное дерево, построенное на сети.

Сетевые модели.
Нахождение потока наименьшей стоимости.
Пример (Х.Таха).

- Компания “Зернышко” снабжает зерном из трех зернохранилищ (узлы 1,2,3) три птицеводческие фермы (узлы 4,5,6). Предложение зернохранилищ составляет 100, 200 и 50 тонн зерна в месяц. Компания может транспортировать зерно по железной дороге, за исключением трех маршрутов, где используется автомобильный транспорт.



- Пропускная способность железных дорог не ограничена. Пропускная способность автотранспорта ограничена снизу и сверху.

Сетевые модели.

Нахождение потока наименьшей стоимости.

Сетевая модель как задача линейного программирования

Для этой задачи можно записать задачу ЛП следующим образом:

min $15x_{12}+20x_{13}+5x_{14}+25x_{23}+30x_{25}+5x_{34}+10x_{35}+10x_{46}+20x_{56}$

$x_{12}+x_{13}+x_{14}=100$

$x_{23}+x_{25}-x_{12}=200$

$x_{34}+x_{35}-x_{13}-x_{23}=50$

$x_{46}-x_{14}-x_{34}=-150$

$x_{56}-x_{25}-x_{35}=-80$

$-x_{46}-x_{56}=-120$

Условие сбалансированности сети: $\sum f_i=0$.

$x_{12} \geq 0; \quad x_{13} \geq 0; \quad x_{14} \leq 80; \quad x_{14} \geq 20$

$x_{23} \geq 0; \quad x_{25} \geq 0;$

$x_{34} \leq 120; \quad x_{34} \geq 20; \quad x_{35} \geq 0;$

$x_{46} \leq 120; \quad x_{46} \geq 10$

$x_{56} \geq 0;$

Сетевые модели.
Нахождение потока наименьшей стоимости.
Пример (Х.Таха).

- Минимальная стоимость – 9050 долларов