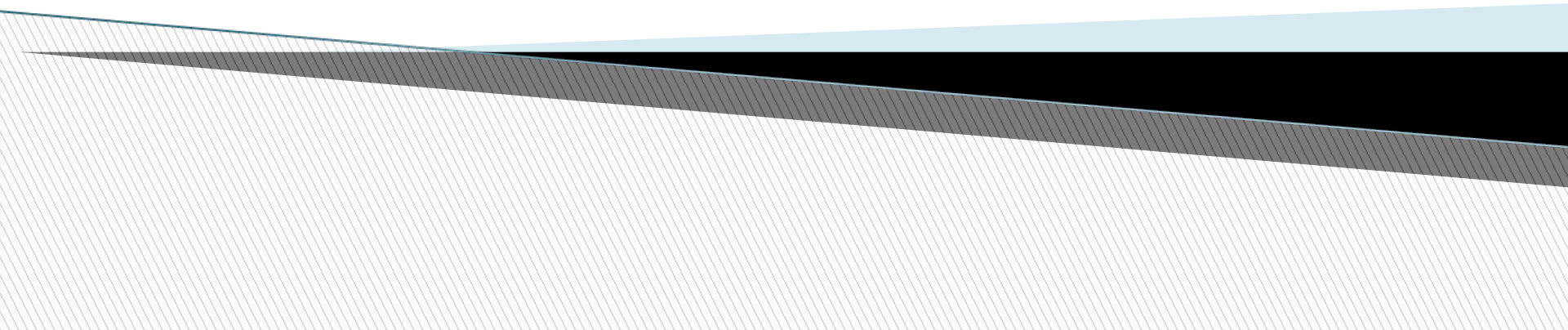


# Сетевое планирование

Элементы теории графов в управлении

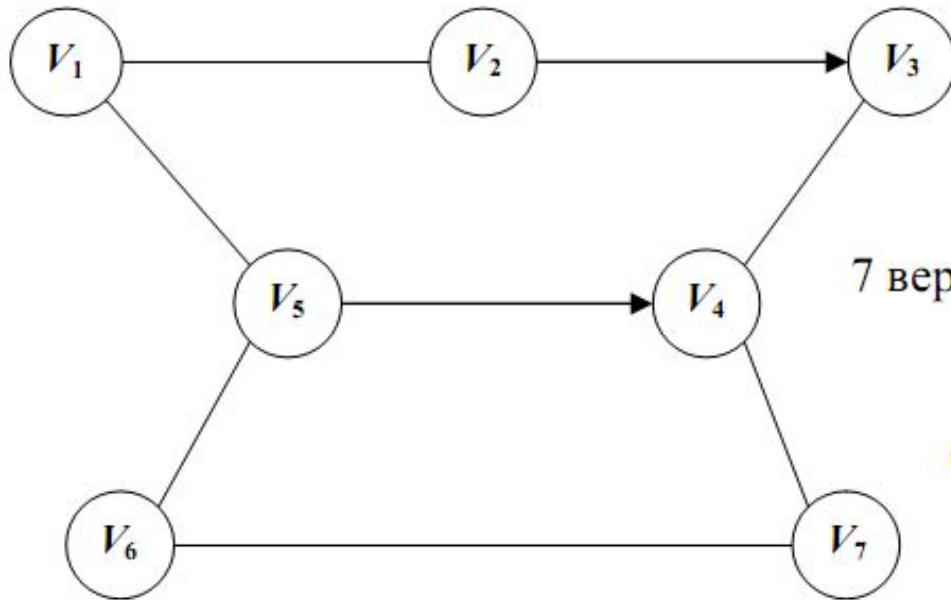


# Основные понятия теории графов

Теория *графов* изучает математические объекты, которые можно изображать в виде рисунков, содержащих кружки и соединяющие их линии.

- Кружки называют *вершинами (узлами)* графа, а соединяющие линии – *ребрами* графа.
- Две вершины графа, соединенные ребром, называют *смежными вершинами*.
- Две смежные вершины называют *граничными вершинами* соединяющего их ребра.
- Ребро, соединяющее две смежные вершины, называют ребром, *инцидентным* этим вершинам.
- Число ребер, инцидентных вершине  $V_i$  графа, называют *степенью вершины*  $V_i$  и обозначают символом  $\deg V_i$ .

# Основные понятия теории графов



7 вершин  $V_1, \dots, V_7$  и 8 ребер  $[V_1, V_2], \dots, [V_6, V_7]$

$\deg V_i = 2$  ( $i = 1, 2, 3, 6, 7$ ),  $\deg V_4 = \deg V_5 = 3$ .

# Основные понятия теории графов

- Ребро графа называют *дугой*, если на нем *при помощи стрелки* указано направление.
- Граничную вершину дуги, из которой стрелка выходит, называют *началом* дуги, а вершина, в которую стрелка входит, – *концом* дуги.
- Граф, в котором *каждое* ребро имеет направление (т.е. является дугой), называют *ориентированным графом*. В противном случае граф называют *неориентированным*.

# Основные понятия теории графов

- Последовательность смежных вершин графа и связывающих их ребер, называют *цепью* (*путём, контуром*).

Каждая цепь состоит из *звеньев*.

- *Звеном* называют две последовательные вершины цепи и связывающее их ребро.

- *Простой цепью* называют цепь, все вершины которой различны.

- *Связным* называют граф, у которого две любые вершины связаны цепью.

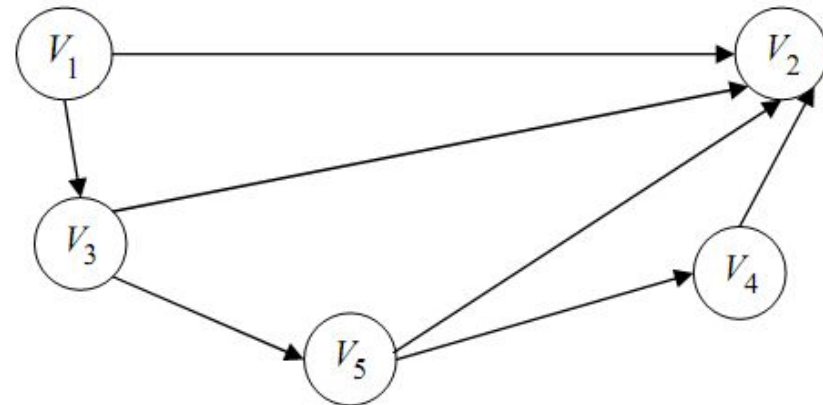
- Замкнутую цепь называют *циклом*.

- Простую замкнутую цепь называют *простым циклом*.

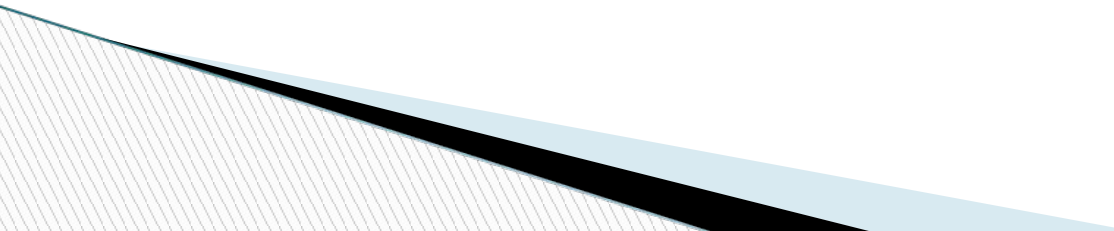
# Понятие сети

*Сетью* называют связный ориентированный граф без ориентированных циклов, удовлетворяющий следующим условиям:

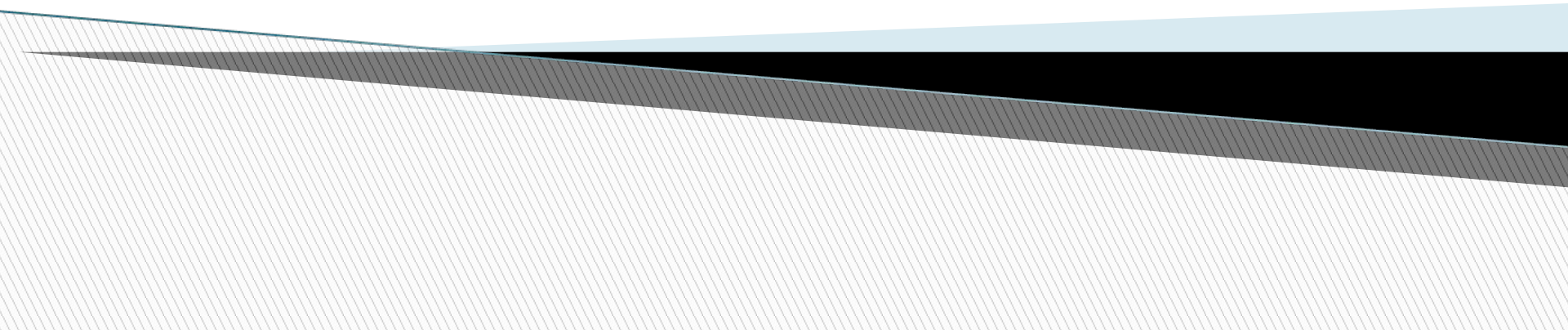
1. Граф имеет единственную вершину, из которой стрелки *выходят*. Эту вершину называют *источником*.
2. Граф имеет единственную вершину, в которую стрелки *входят*. Эту вершину называют *стоком*.
3. Все остальные вершины имеют как входящие, так и выходящие стрелки.
4. Две смежные вершины соединяет только одна стрелка.



# Сетевое планирование

- Сети в экономике и управлении используют для графического отображения, планирования и последующего управления **технологическим комплексом** – системой взаимосвязанных операций и работ.
  
  - При этом выделяют три этапа сетевого планирования:
    1. **Структурное планирование** – осознание взаимосвязи отдельных операций и отображение их в виде сети;
    2. **Календарное планирование** – расчет временных параметров сети;
    3. **Оперативное управление** – осуществление управления на основе решенной задачи с внесением необходимых поправок, вызванных реальным течением процесса (относится к менеджменту)
- 

# Структурное планирование





# Понятие работы в сети

Построение сетевой модели (**структурное планирование**) начинается с разбиения проекта на четко определенные работы, для которых определяется продолжительность.

**Работа** – это некоторый процесс, приводящий к достижению определенного результата, требующий затрат каких-либо ресурсов и имеющий протяженность во времени. По количеству затрачиваемого времени работа может быть:

- **действительной**, т.е. требующей затрат времени;
- **фиктивной**, т.е. формально не требующей затрат времени.

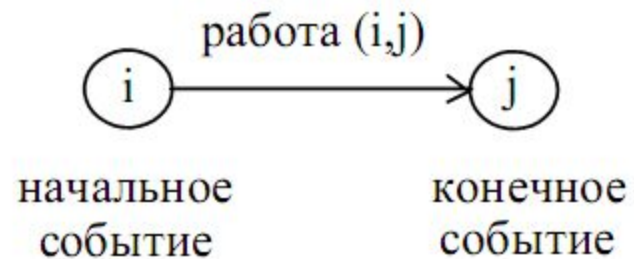
# Понятие события в сети

Работы связаны друг с другом таким образом, что выполнение одних работ может быть начато *только после* завершения некоторых других. **Событие** – это момент времени, когда завершаются одни работы и начинаются другие. Событие представляет собой результат проведенных работ и, в отличие от работ, не имеет протяженности во времени.

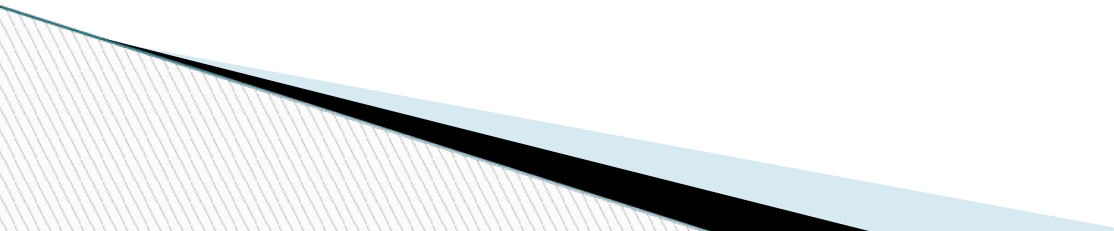
Событие, не имеющее предшествующих ему событий, т.е. с которого начинается проект, называют **исходным**. Событие, которое не имеет последующих событий и отражает конечную цель проекта, называется **завершающим**.

# Понятие сетевого графика

Взаимосвязь работ и событий, необходимых для достижения конечной цели проекта, изображается с помощью **сетевого графика** (сетевой модели). Работы изображаются *стрелками*, которые соединяют *вершины*, изображающие события. Начало и окончание любой работы описываются парой событий, которые называются **начальным** и **конечным** событиями. Поэтому для указания конкретной работы используют код работы  $(i, j)$ , состоящий из номеров начального ( $i$ -го) и конечного ( $j$ -го) событий



Исходные данные для построения сетевой модели могут задаваться различными способами, например,

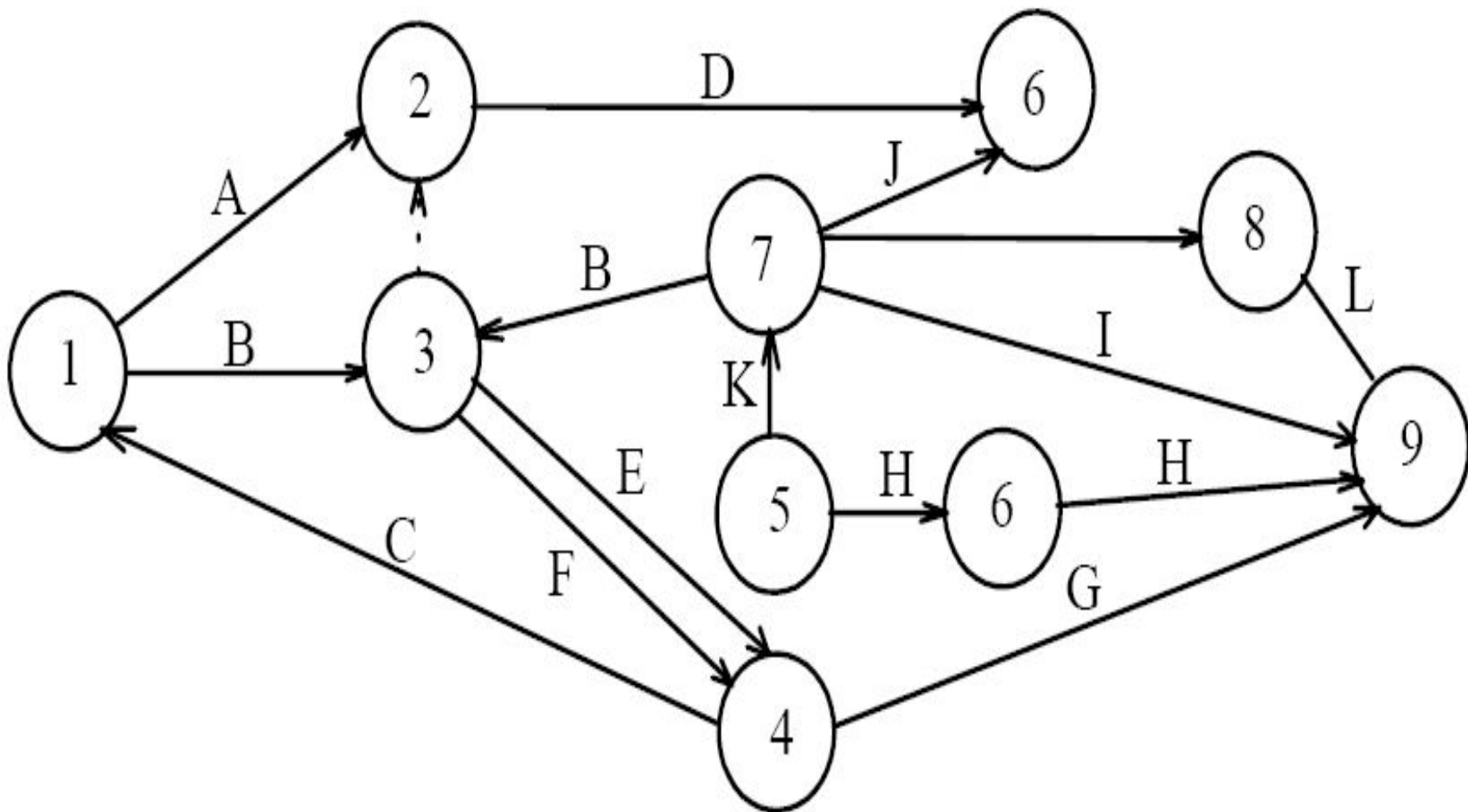
- описанием предполагаемого проекта. В этом случае необходимо самостоятельно разбить его на отдельные работы и установить их взаимные связи;
  - списком работ проекта. В этом случае необходимо проанализировать содержание работ и установить существующие между ними связи;
  - списком работ проекта с указанием их упорядочения. В этом случае необходимо только отобразить работы на сетевом графике.
- 

# Правила построения сети

- длина стрелки не зависит от времени выполнения работы;
- стрелка может не быть прямолинейным отрезком;
- для действительных работ используются сплошные, а для фиктивных – пунктирные стрелки;
- каждая операция должна быть представлена только одной стрелкой;
- между одними и теми же событиями не должно быть **параллельных** работ, т.е. работ с одинаковыми кодами;
  - не должно быть **висячих** событий (т.е. не имеющих предшествующих событий), кроме исходного;
  - не должно быть **тупиковых** событий (т.е. не имеющих последующих событий), кроме завершающего;
    - не должно быть стрелок, направленных справа налево;
    - не должно быть циклов

# Задача 1.

Найдите ошибки в построении сети

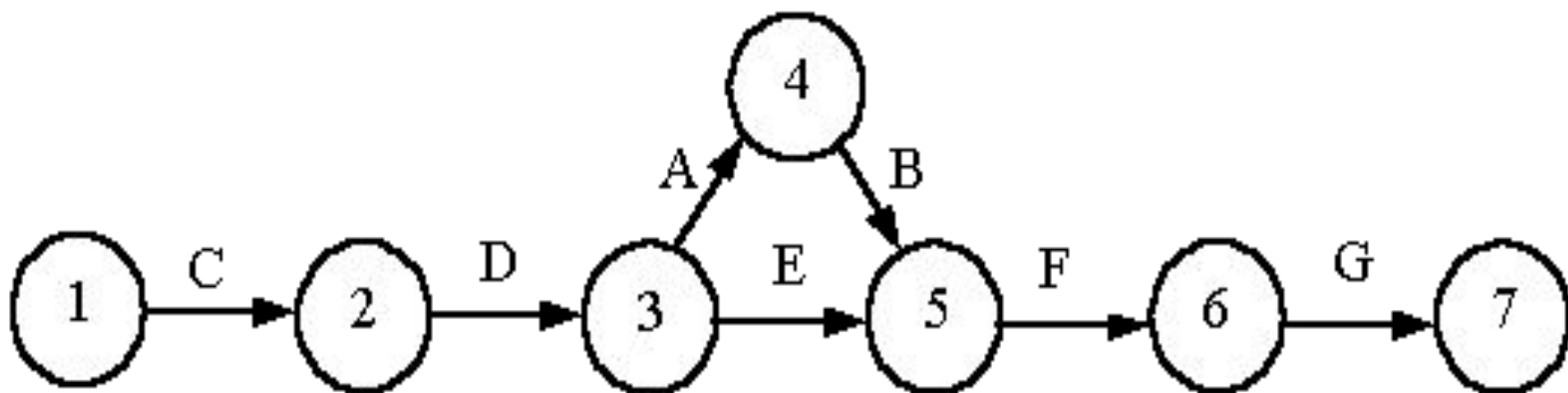


# Задача №2

Постройте сетевую модель программы опроса общественного мнения, которая включает:

- разработку анкет (А; 1 день),
- распечатку анкет (В; 0,5 дня),
- прием на работу персонала (С; 2 дня),
- обучение (D; 2 дня) персонала,
- выбор опрашиваемых лиц (Е; 2 дня),
- рассылку анкет опрашиваемым (F; 1 день),
- анализ полученных данных (G; 5 дней).

# Сетевая модель к задаче №2



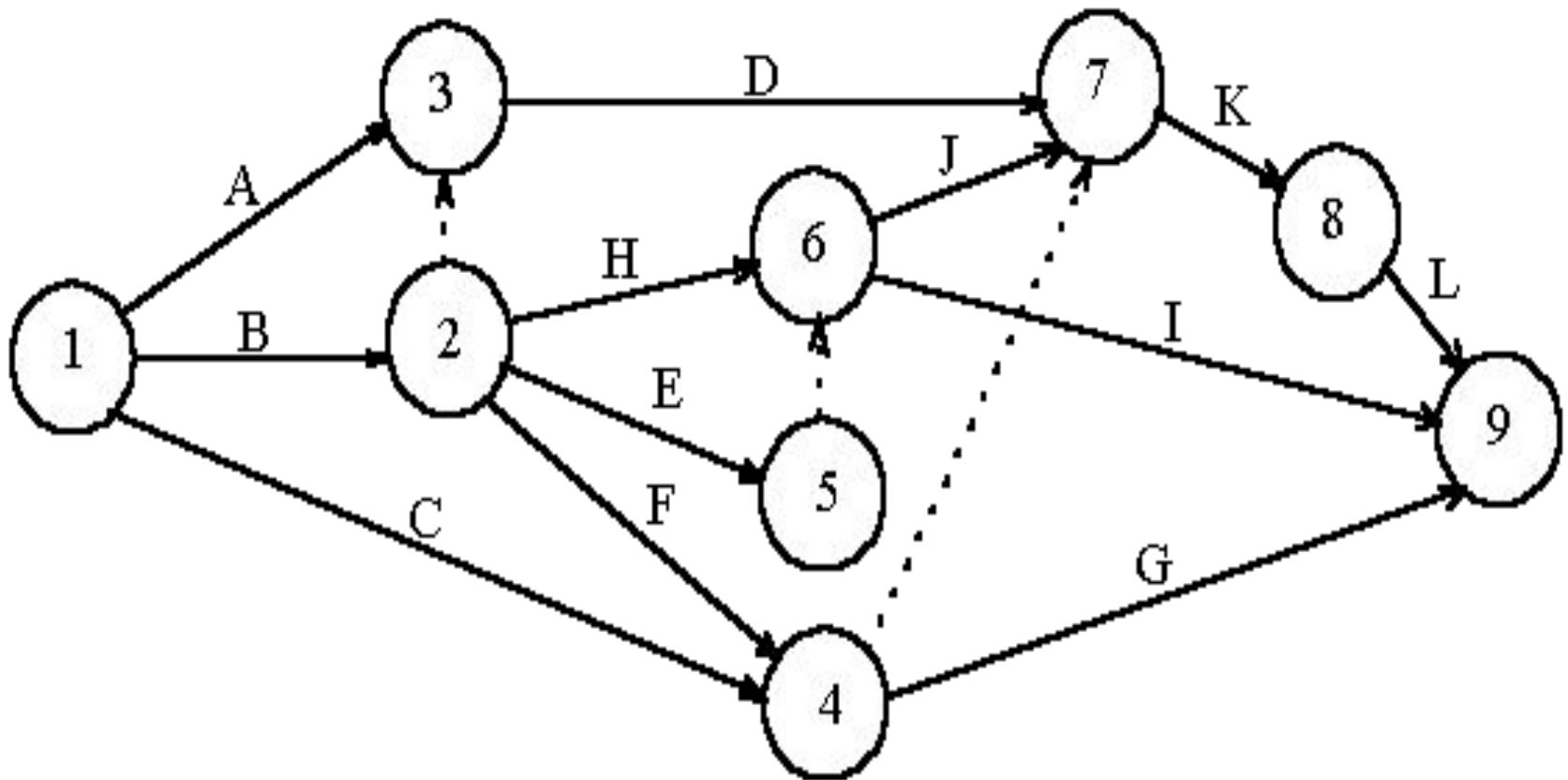


# Задача №3

Постройте сетевую модель, включающую работы А, В, С, ..., L, которая отображает следующее упорядочение работ:

- 1) А, В и С – исходные операции проекта;
- 2) А и В предшествуют D;
- 3) В предшествует Е, F и H;
- 4) F и С предшествует G;
- 5) Е и H предшествуют I и J;
- 6) С, D, F и J предшествуют K;
- 7) K предшествует L.

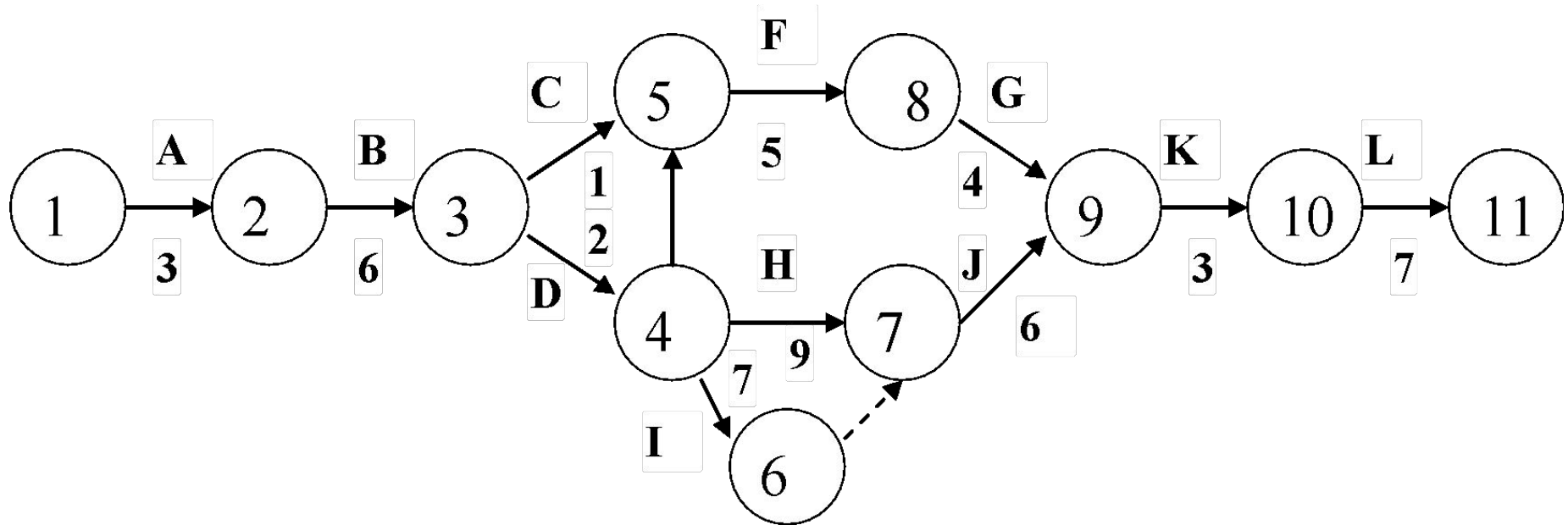
# Сетевая модель к задаче №3



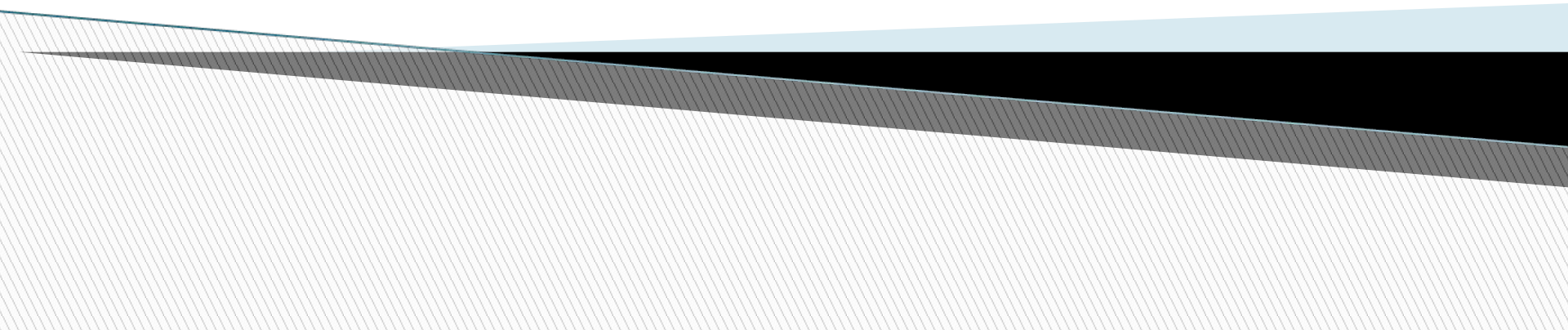
## Задача №4. Построить сетевой график, используя упорядочение работ из таблицы

Работа	Непосредственно предшествующие работы	Время, ед. времени
А – составление сметы затрат	–	3
В – согласование оценок	А	6
С – покупка собственного оборудования	В	1
Д – подготовка конструкторских проектов	В	2
Е – строительство основного цеха	Д	1
Ф – монтаж оборудования	С,Е	5
Г – испытание оборудования	Ф	4
Н – определение типа модели	Д	9
І – проектирование внешнего корпуса	Д	7
Ј – создание внешнего корпуса	Н,І	6
К – конечная сборка	Г,Ј	3
Л – контрольная проверка	К	7

# Сетевой график к задаче №4.



# Календарное планирование

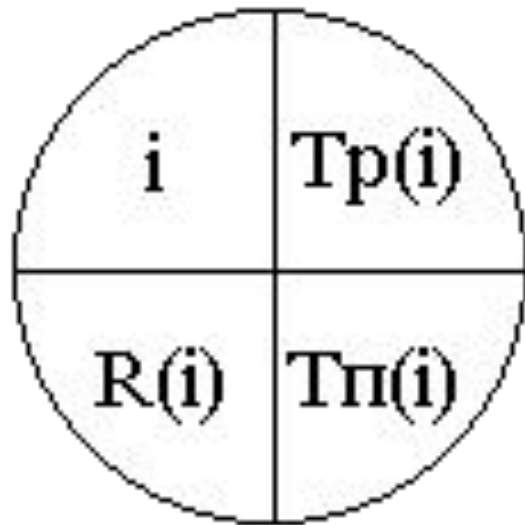


# Задачи календарного планирования

**Календарное планирование** предусматривает определение моментов начала и окончания каждой работы и других временных характеристик сетевого графика. Это позволяет проанализировать сетевую модель, выявить критические работы, непосредственно определяющие срок выполнения проекта, провести оптимизацию использования ресурсов (временных, финансовых, исполнителей).

*Расчет сетевой модели начинают с временных параметров событий, которые вписывают непосредственно в вершины сетевого графика*

# Отображение временных параметров событий на сетевом графике



$i$  – номер события

$T_p(i)$  – ранний срок наступления события  $i$ , минимально необходимый для выполнения всех работ, которые предшествуют событию  $i$ ;

$T_{п}(i)$  – поздний срок наступления события  $i$ , превышение которого вызовет аналогичную задержку наступления завершающего события сети;

$R(i)$  – резерв события  $i$ , т.е. время, на которое может быть отсрочено наступление события  $i$  без нарушения сроков завершения проекта в целом.

# Расчет ранних сроков свершения событий

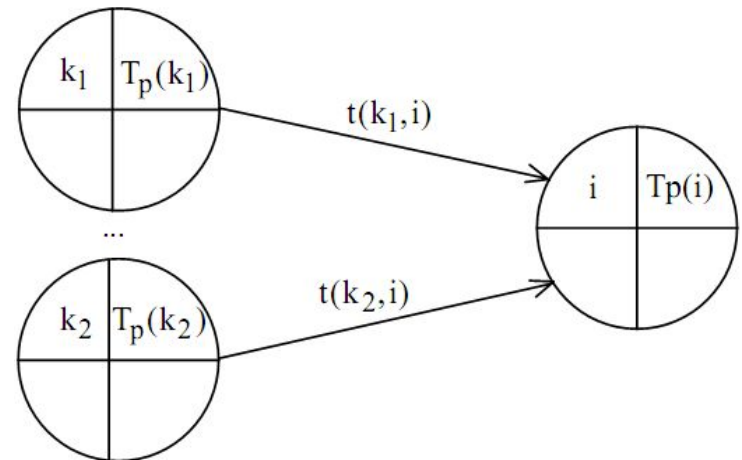
Ранние сроки свершения событий  $T_p(i)$  рассчитываются от исходного (И) к завершающему (З) событию следующим образом:

к завершающему (З) событию следующим образом:

- 1) для исходного события И  $T_p(И) = 0$ ;
- 2) для всех остальных событий I

$$T_p(i) = \max_{\forall(k,i)} [T_p(k) + t(k,i)],$$

где максимум берется по всем работам  $(k,i)$ , входящим в событие  $i$ ;  $t(k,i)$  – длительность работы  $(k,i)$





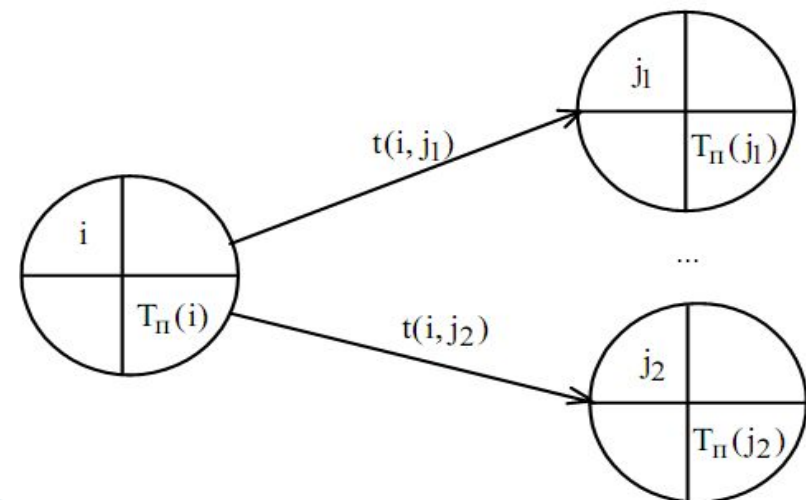
# Расчет поздних сроков свершения событий

Поздние сроки свершения событий  $T_{п}(i)$  рассчитываются от завершающего к исходному событию:

- 1) для завершающего события 3  $T_{п}(3) = T_{р}(3)$ ;
- 2) для всех остальных событий

$$T_{п}(i) = \min_{\forall(i,j)} [T_{п}(j) - t(i,j)],$$

где минимум берется по всем работам  $(i,j)$ , выходящим из события  $i$ ;  $t(k,i)$  – длительность работы  $(k,i)$  (рис. 8.3).



# Временные параметры работ

Временные параметры работ определяются на основе ранних и поздних сроков событий:

- $T_{рн}(i, j) = T_p(i)$  – ранний срок начала работы;
- $T_{ро}(i, j) = T_p(i) + t(i, j)$  – ранний срок окончания работы;
- $T_{по}(i, j) = T_{п}(j)$  – поздний срок окончания работы;
- $T_{пн}(i, j) = T_{п}(j) - t(i, j)$  – поздний срок начала работы;
- $R_{п}(i, j) = T_{п}(j) - T_p(i) - t(i, j)$  – полный резерв работы показывает

максимальное время, на которое можно увеличить длительность работы  $(i, j)$  или отсрочить ее начало, чтобы не нарушился срок завершения проекта в целом;

- $R_c(i, j) = T_p(j) - T_p(i) - t(i, j)$  – свободный резерв работы показывает максимальное время, на которое можно увеличить продолжительность работы  $(i, j)$  или отсрочить ее начало, не меняя ранних сроков начала последующих работ.

# Пути в сети

**Путь** – это последовательность работ в сетевом графике (в частном случае это одна работа), в которой конечное событие одной работы совпадает с начальным событием следующей за ней работы.

**Полный путь** – это путь от исходного до завершающего события.

**Критический путь** – максимальный по продолжительности полный путь.

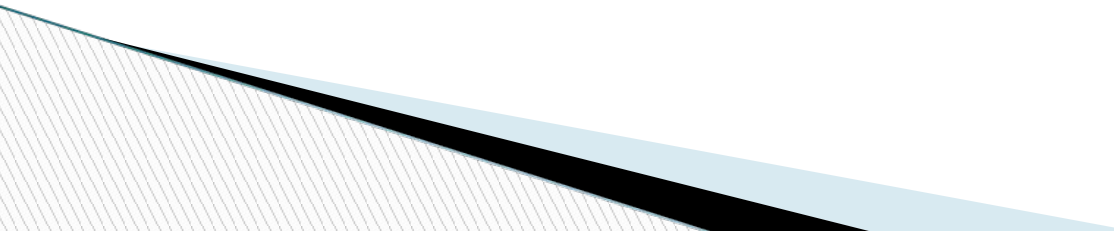
Работы, лежащие на критическом пути, называют **критическими**.

Критические работы имеют нулевые свободные и полные резервы.

**Подкритический путь** – полный путь, ближайший по продолжительности к критическому пути.

# Критический путь в сети

При поиске критических путей на сетевом графике будем использовать следующие условия его критичности:

- *необходимое условие – нулевые резервы событий, лежащих на критическом пути;*
  - *достаточное условие – нулевые полные резервы работ, лежащих на критическом пути.*
- 

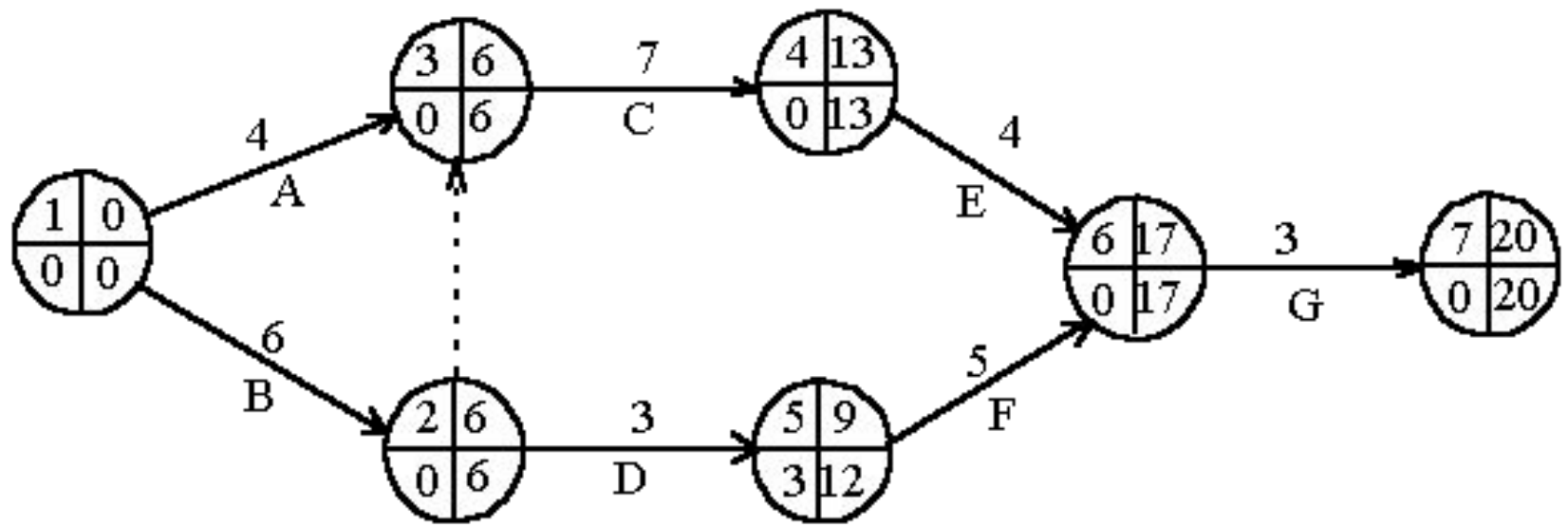
# Задача №5

Компания разрабатывает строительный проект. Исходные данные по операциям проекта представлены в табл.

Постройте сетевую модель проекта, определите критические пути модели и проанализируйте, как влияет на ход выполнения проекта задержка работы D на 4 недели.

Название	Предшествующие операции	Длительность, нед.
A	-	4
B	-	6
C	A,B	7
D	B	3
E	C	4
F	D	5
G	E,F	3

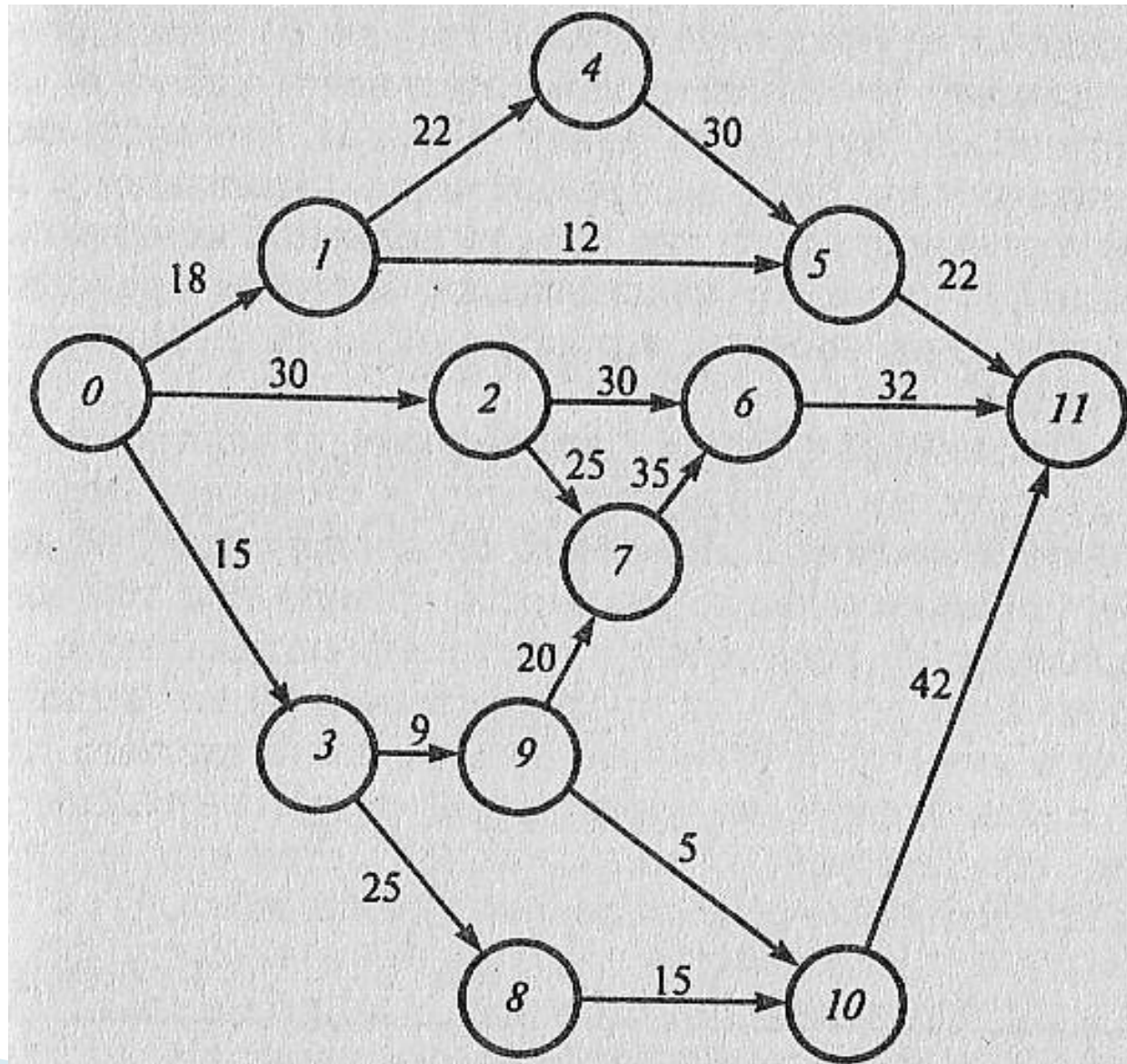
# Сетевой график к задаче №5



# Задача №6.

Для сетевого графика найти

- все полные пути,
- критический путь,
- ранние и поздние сроки свершения событий, полные и свободные резервы работ



# График привязки

Для проведения анализа временных параметров сетевой модели используют **график привязки**, который отображает взаимосвязь выполняемых работ во времени. По вертикальной оси графика привязки откладываются коды работ, по горизонтальной оси – отрезки, соответствующие длительностям работ (раннее начало и раннее окончание работ). График привязки можно построить на основе данных о продолжительности работ. При этом необходимо помнить, что работа  $(i, j)$  может выполняться только после того как будут выполнены все предшествующие ей работы  $(k, i)$ .



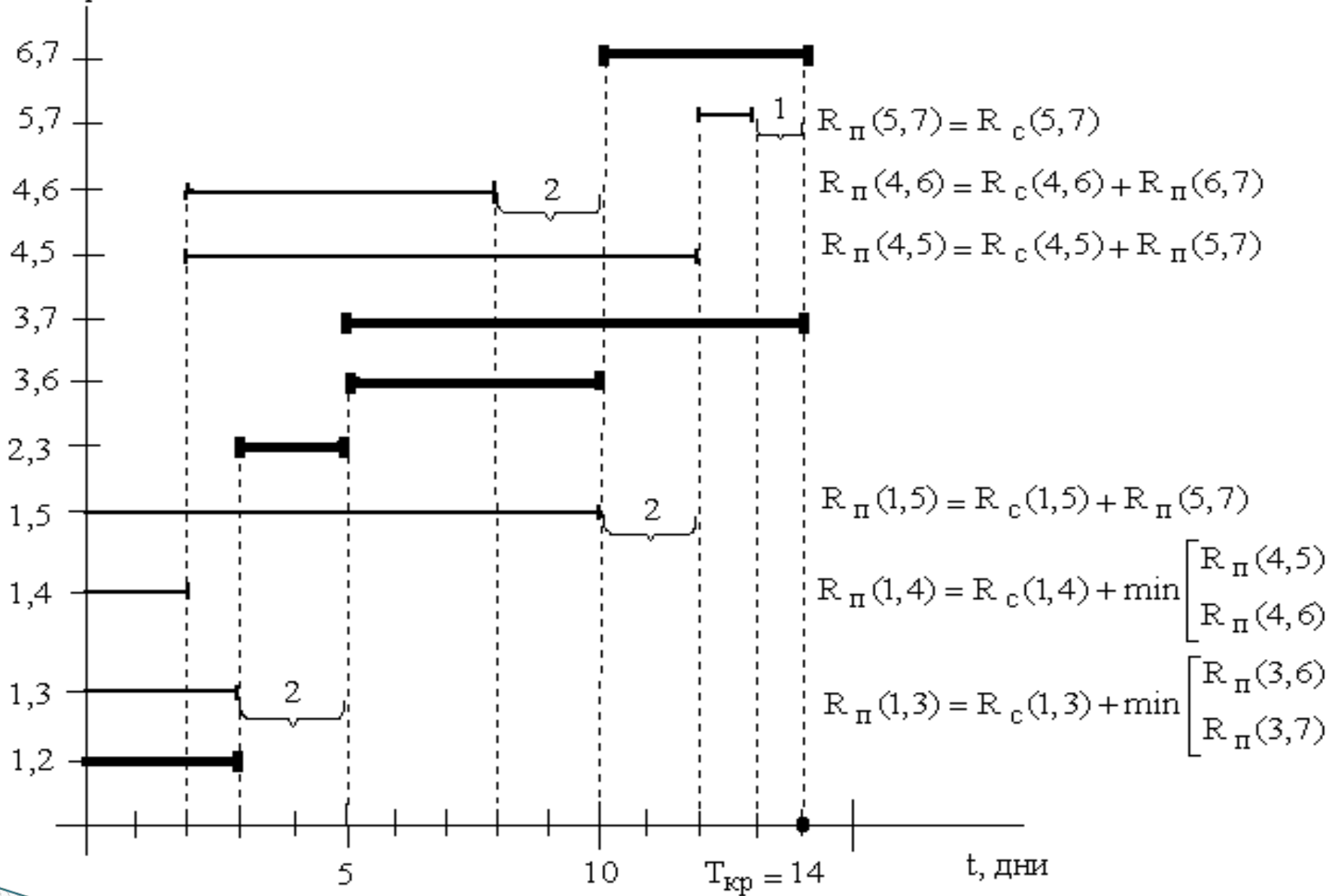
# Задача №7.

По данным о кодах и длительностях работ в днях (см. табл.) постройте **график привязки** сетевой модели, определите критические пути и их длительность.

Определите свободные и полные резервы каждой работы, отметьте на графике привязки свободные резервы работ.

<b>(i,j)</b>	<b>1,2</b>	<b>1,3</b>	<b>1,4</b>	<b>1,5</b>	<b>2,3</b>	<b>3,6</b>	<b>3,7</b>	<b>4,5</b>	<b>4,6</b>	<b>5,7</b>	<b>6,7</b>
<b>T(i,j)</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>10</b>	<b>2</b>	<b>5</b>	<b>9</b>	<b>10</b>	<b>6</b>	<b>1</b>	<b>4</b>

Код работы



**График привязки к задаче №7**