

# \* Сетевые графики

**Сетевой график** - информационно-динамическая модель, отражающая взаимосвязи между работами, необходимые для достижения конечной цели проекта.

Работами являются любые действия, приводящие к достижению определенных результатов - **событий**.

События, кроме исходного, являются результатами выполнения работ.

Между двумя смежными событиями может выполняться только одна работа или последовательность работ.

**Путь** - последовательность взаимосвязанных работ, ведущая из одной вершины проекта в другую вершину.

**Длина пути** - суммарная продолжительность выполнения всех работ пути.

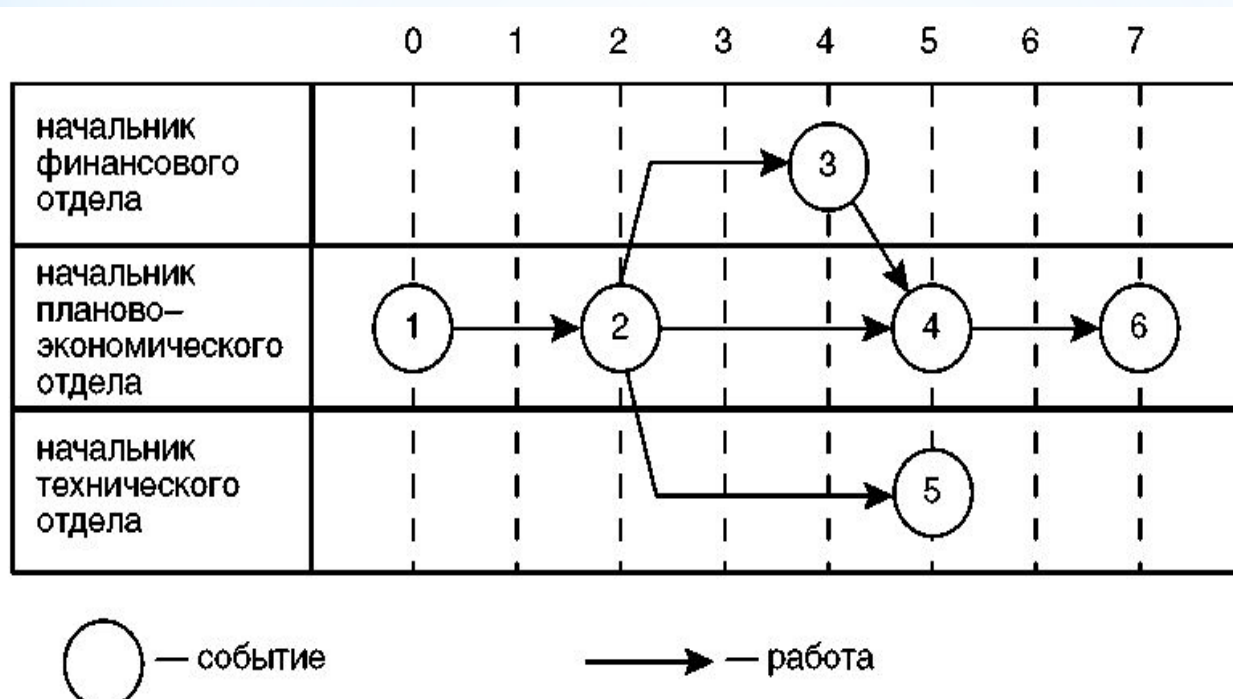
**Полный путь** - это путь от исходного к завершающему событию.

**Критический путь** - полный путь, суммарная продолжительность выполнения всех работ которого является наибольшей.

## 2 подхода к построению сетевых моделей.

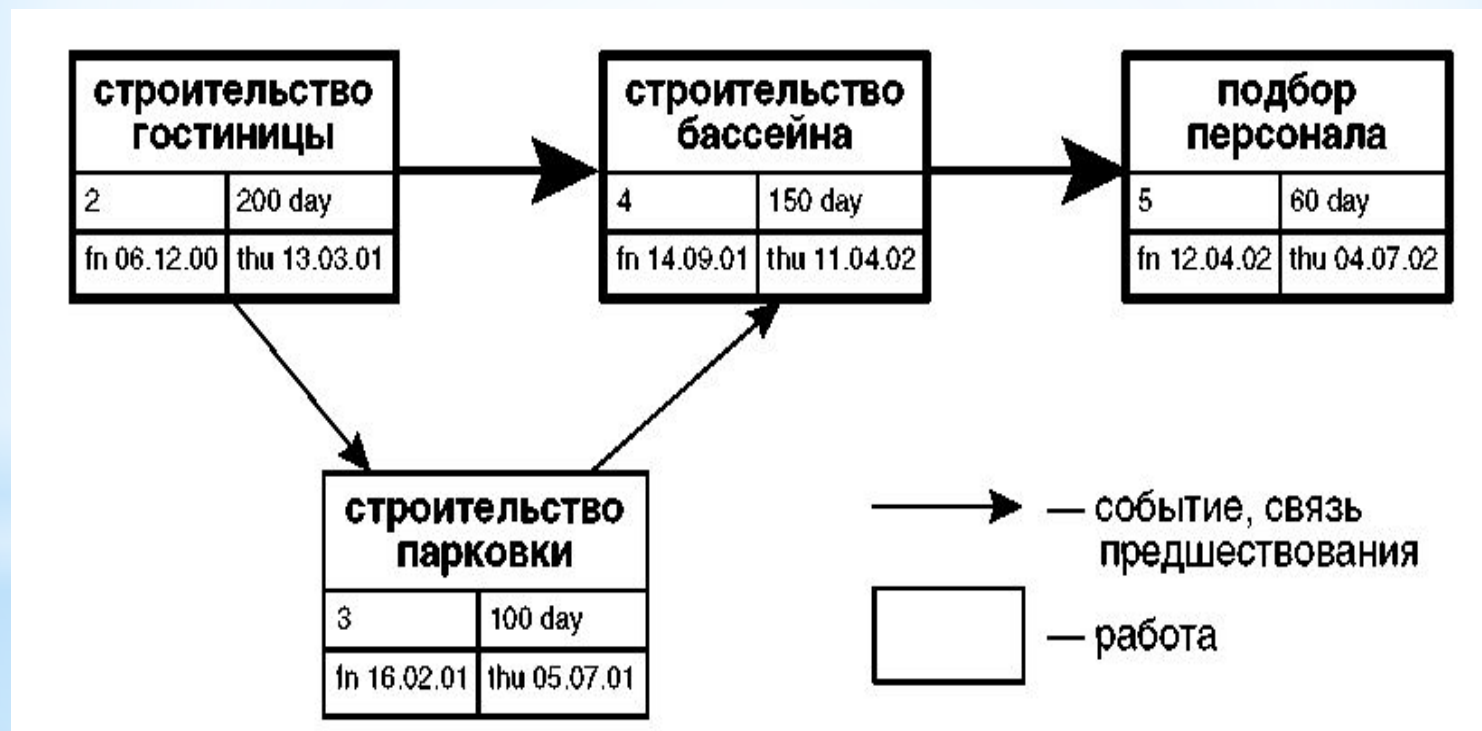
### 1. «Работа-стрелка», сетевые графики:

стрелками на графике изображаются работы, а вершинами - события.



## 2. «Работа-вершина», сети предшествования

Стрелкам соответствуют события, а вершинам - работы.



Для построения сетевых моделей необходимо определить логические взаимосвязи между работами.

Причиной взаимосвязей являются, как правило, технологические ограничения (начало одних работ зависит от завершения других).

Комплекс взаимосвязей между работами определяет последовательность выполнения работ во времени.

Для определения последовательности действий необходима следующая информация: перечень работ, описание продукта, технологические, взаимосвязи, ограничения и предположения.



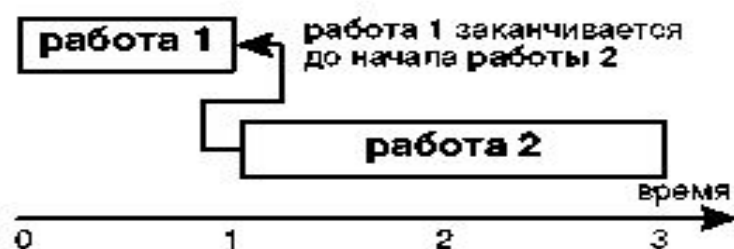
P2  
начнется  
после  
окончания  
P1



P2  
начнется  
если  
начнется  
P1



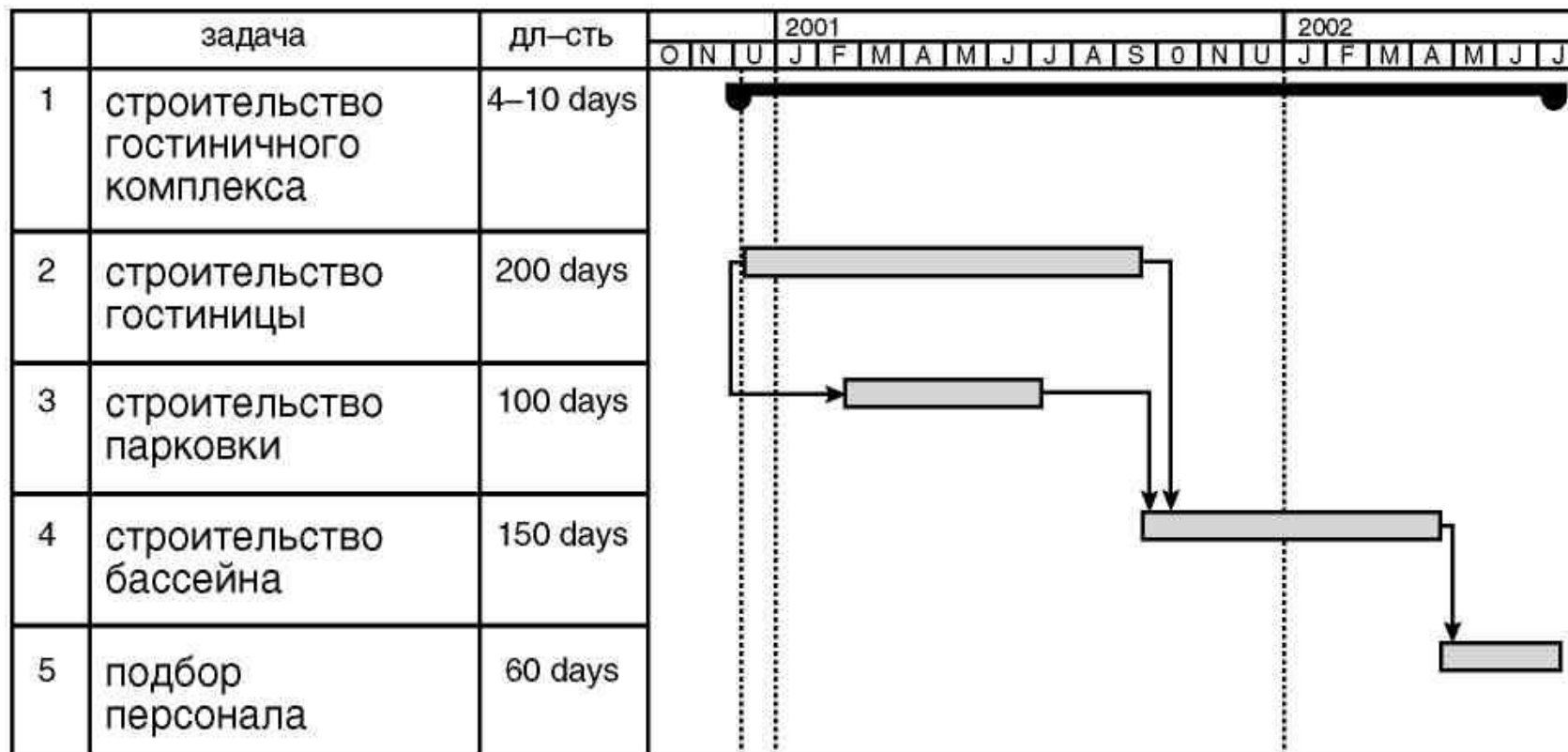
P2  
закончится  
если  
закончится  
P1



P1  
должна  
закончиться  
до начала  
P2



# Диаграмма Гантта



## Разработка идеального календарного графика работ

**Цель:** минимизация временных затрат на выполнение работ за счет проведения оценки длительности выполнения по каждой из них.

При определении идеальных оценок длительности работ пренебрегают ограничениями ресурсов и учитывают только технологические ограничения и нормативы.

Результаты определения длительности работ вводят в сетевую модель и определяют общую продолжительность проекта.

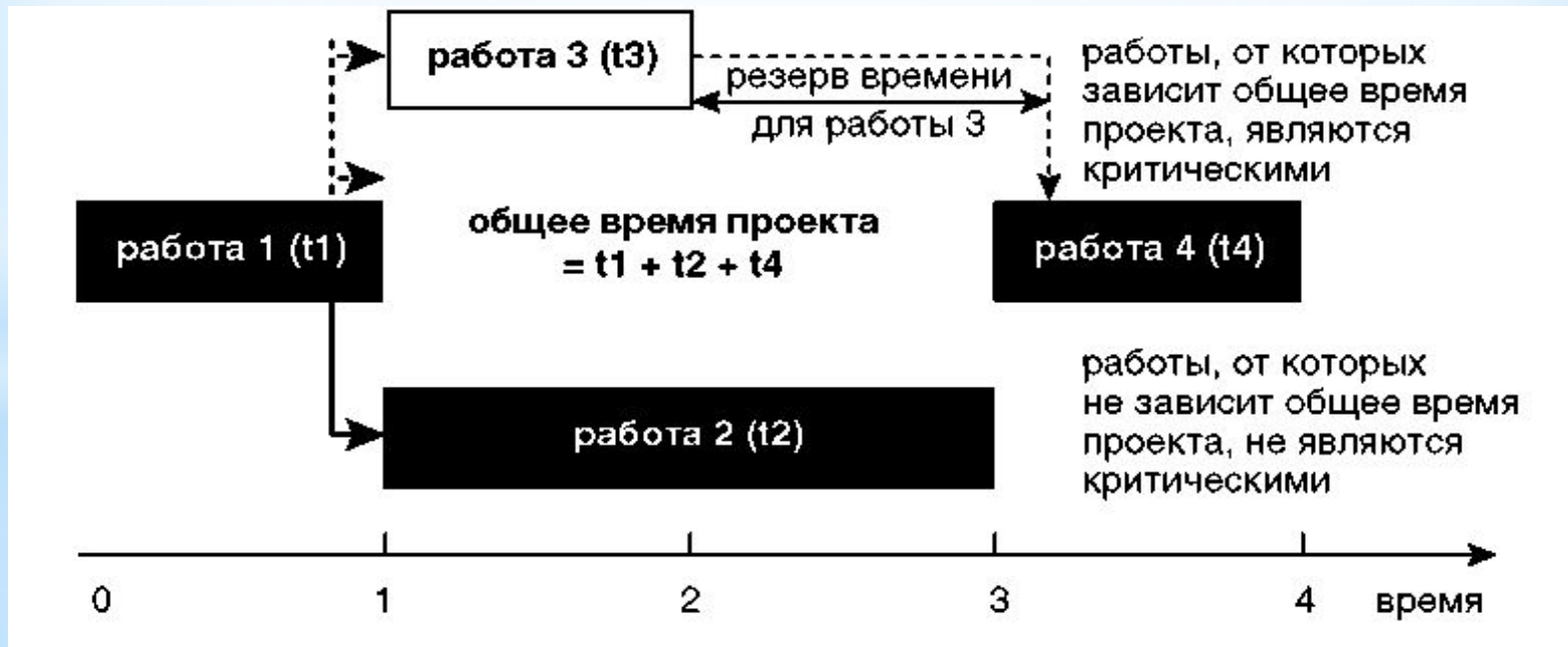
Уменьшение общей длительности проекта возможно за счет:

- сокращения времени выполнения отдельных работ
- организации их параллельного выполнения
- устранения временных разрывов.

Для выделения работ, определяющих минимальную длительность проекта, применяется метод критического пути.

Для временной оптимизации сетевой модели необходимо определить именно те работы, которые в действительности определяют его длительность, и отделить их от тех, которые не влияют на суммарное время реализации проекта.

Такие работы называют **критическими**, а их сетевую взаимосвязь (цепочку) - **критическим путем**.



## Методика определения критического пути

**Критический путь** - наиболее протяженная по времени цепочка работ, ведущая от исходного к завершающему событию.

Изменение продолжительности любой работы, лежащей на критическом пути, соответственным образом меняет (сокращает или удлиняет) срок наступления завершающего события, т.е. дату завершения проекта, поскольку работы, лежащие на критическом пути не имеют резерва времени.

Все работы, которые лежат вне критического пути, имеют резерв времени, на которое может быть отсрочено наступление завершения данной работы без нарушения сроков проекта в целом.

Резерв времени работы определяется как разность между поздним и ранним сроками завершения работы.

Ранний из возможных сроков наступления окончания работы - это срок, необходимый для выполнения всех работ, предшествующих данной. Поздний из допустимых сроков - это такой промежуток времени, на который может быть отсрочено наступление окончания данной работы без нарушения сроков проекта в целом.

Все работы лежащие вне не критического пути обладают резервами времени в рамках которых их можно сдвигать, не приводя к изменению времени выполнения всего проекта

## Методика определения критического пути

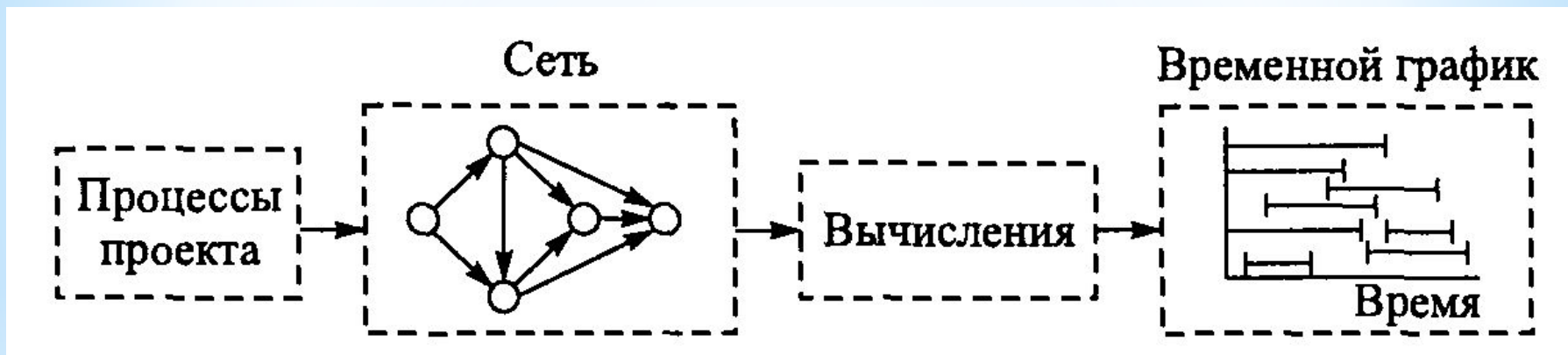
**Прямой проход** начинается с начальной даты проекта и продолжается по сетевому графику, при этом для каждой работы к начальному сроку прибавляется ее продолжительность и вычисляется **раннее начало** и **раннее окончание** этой работы. Самыми ранними возможными датами для работы являются ее начало и конец, допустимые исходя из логической последовательности сетевого графика.

**Обратный проход** использует в качестве исходной точки конечную дату расписания, вычисленную путем прямого прохода, и проводится обратный расчет для определения поздних дат начала и окончания работ. Поздними датами работы являются ее самые **поздние допустимые начало** и **конец**, не влекущие задержки выполнения всего проекта.

Кроме того, на основании рассчитанных ранних и поздних дат начала работ определяются величины **временных резервов** для каждой работы.

## Разработка реального календарного графика работ

- \* Практический смысл этого шага проектирования состоит в том, чтобы учесть все необходимые ресурсы, правильно оценить и распределить их.
- \* Ресурсами, планируемыми в проекте, являются ресурсы материальные и человеческие. Все они имеют стоимостную оценку, однако ее недостаточно для планирования, важно оценить такие параметры как наличие ресурсов и их доступность, качество, сезонность и многое другое.

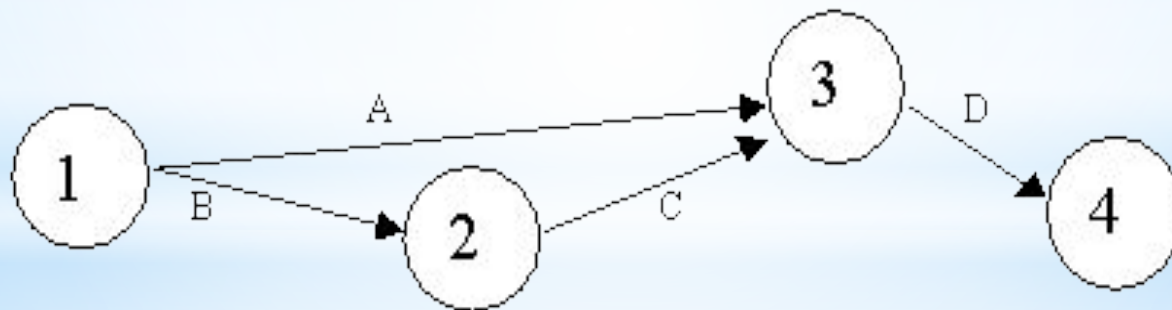




## Этапы сетевого планирования:

- 1.** Разбиение проекта на четко определенные работы, для которых определяется продолжительность.
- 2.** Построение сетевого графика, показывающей последовательность работ, составляющих проект.
- 3.** На третьем этапе на основе построенной сети выполняются вычисления, в результате которых составляется временной график реализации проекта

| Работа | Непосредственно предшествующая работа | Время выполнения |
|--------|---------------------------------------|------------------|
| A      | -                                     | $t_A$            |
| B      | -                                     | $t_B$            |
| C      | B                                     | $t_C$            |
| D      | A, C                                  | $t_D$            |





Каждая вершина графа отображает *событие* (момент времени, когда завершаются одни работы и начинаются другие).

Событие 1 означает начало выполнения проекта.

Событие 4 означает завершение проекта.

Любая работа проекта - это упорядоченная пара двух событий.

Например, работа A есть упорядоченная пара событий (1,3).

Работа D - упорядоченная пара событий (3,4).

Событие проекта состоит в том, что завершены все работы, «входящие» в соответствующую вершину.

Например, событие 3 состоит в том, что завершены работы A и C.

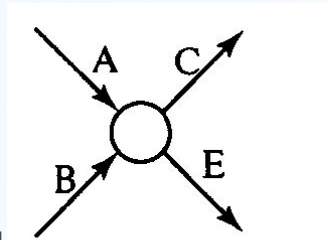
## Построение сети проекта основано на следующих правилах:

1. Каждая работа в проекте представляется одной и только одной дугой.
2. Каждая работа идентифицируется двумя концевыми узлами
3. Для поддержания правильных отношений предшествования при включении в сетевой график любой работы необходимо ответить на следующие вопросы.
  - \* Какая работа непосредственно предшествует текущей?
  - \* Какая работа должна выполняться после завершения текущей работы?
  - \* Какая работа конкурирует (выполняется параллельно) с текущей?

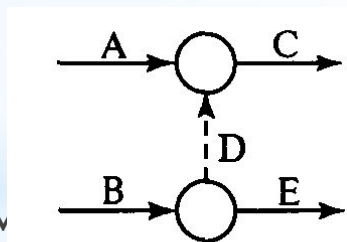
Ответы на эти вопросы, возможно, потребуют включить в сеть **фиктивные работы**, чтобы правильно отобразить последовательность выполнения работ. Предположим, например, что четыре работы должны удовлетворять следующим условиям.

1. Работа С должна начаться сразу после завершения работ А и В.
2. Работа Е должна начаться непосредственно после завершения работы В.

Неправильное представление работ, так как из него следует, что работа Е должна начаться после завершения как работы В, так и А.



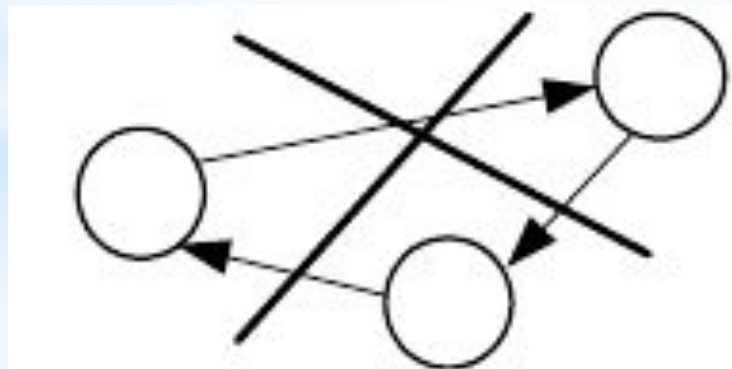
Решение проблемы с помощью фиктивной работы D:



Фиктивная работа может существовать, например, «передача документов от одного отдела к другому». Если продолжительность такой работы несоизмеримо мала по сравнению с продолжительностью других работ проекта, то формально ее принимают равной 0.

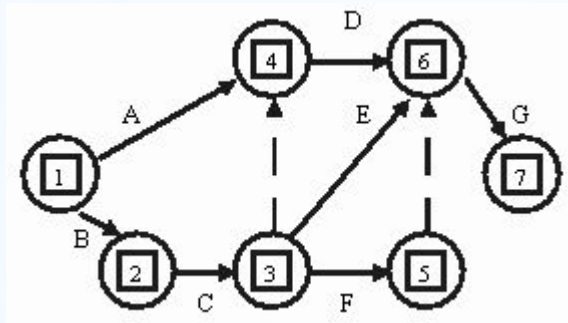
В сетевом графике не должно быть:

- «висячих» событий (т.е. не имеющих предшествующих событий), кроме исходного;
- тупиковых событий (т.е. не имеющих последующих событий), кроме завершающего;
- циклов



## Определение критического пути.

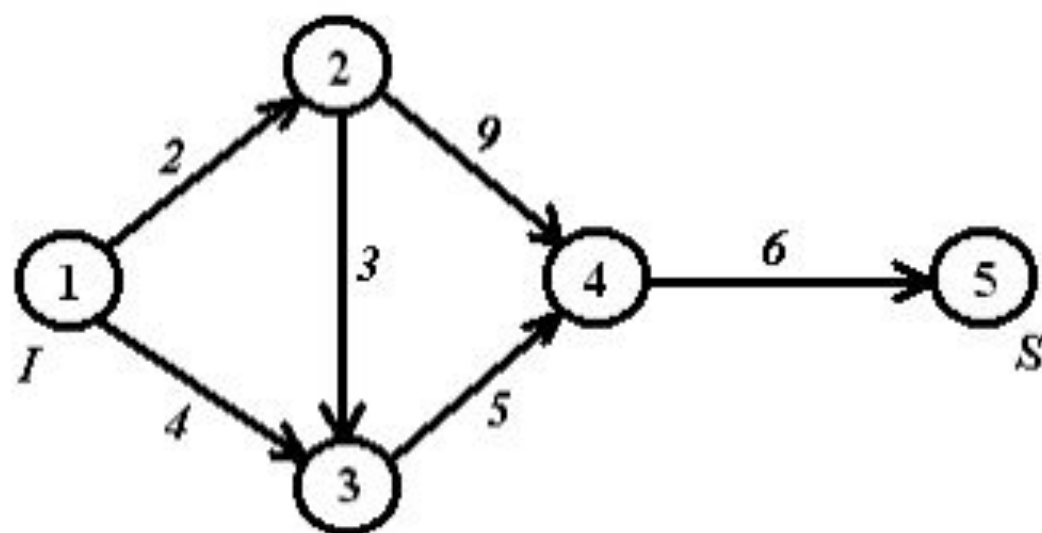
**Путь** - последовательность взаимосвязанных работ, ведущая из одной вершины проекта в другую вершину. Например {A, D, G} и {C, F} - два различных пути.



**Длина пути** - суммарная продолжительность выполнения всех работ пути.

**Полный путь** - это путь от исходного к завершающему событию.

**Критический путь** - полный путь, суммарная продолжительность выполнения всех работ которого является наибольшей.



Полные пути и их продолжительности:

1)  $1 - 2 - 3 - 4 - 5 \Rightarrow 2+3+5+6=16;$

2)  $1 - 2 - 4 - 5 \Rightarrow 2+9+6=17;$

3)  $1 - 3 - 4 - 5 \Rightarrow 4+5+6=15.$

← критический срок (путь)



**Резерв времени событий** показывает, на сколько времени может задержаться свершение этого события без изменения срока наступления завершающего (конечного) события (стока).

Для определения резервов времени событий необходимо рассчитать ранние и поздние сроки свершения событий.

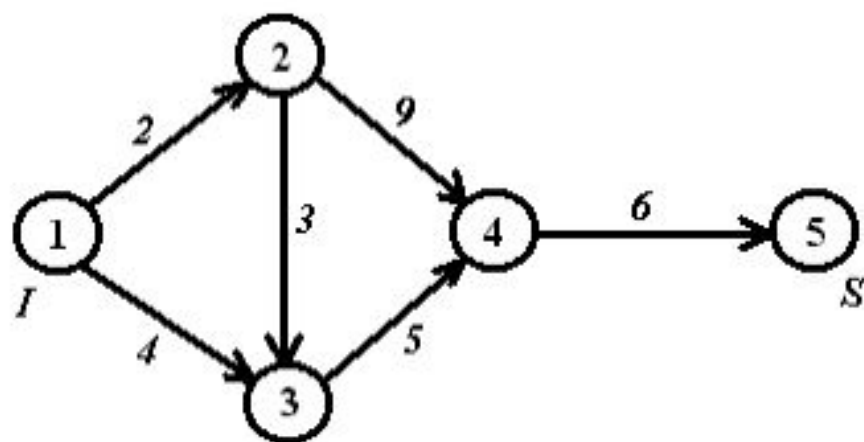
**Ранний срок  $t_p(j)$  свершения события  $j$**  - это ранний срок, необходимый для выполнения всех работ, предшествующих данному событию:

$$t_p(j) = \max_{(i,j) \in U_j^+} (t_p(i) + t(i,j)),$$

где  $t_p(i)$  - ранний срок свершения события  $i$ ;

$t(i,j)$  - продолжительность работы  $(i,j)$ ;

$U_j^+$  - множество работ, входящих в событие  $j$ .



$$t_p(1) = 0;$$

$$t_p(2) = t_p(1) + t(1, 2) = 0 + 2 = 2;$$

$$t_p(3) = \max \{ (t_p(1) + t(1, 3)), (t_p(2) + t(2, 3)) \} = \max \{ (0 + 4), (2 + 3) \} = \\ = \max \{ 4, 5 \} = 5;$$

$$t_p(4) = \max \{ (t_p(2) + t(2, 4)), (t_p(3) + t(3, 4)) \} = \max \{ (2 + 9), (5 + 5) \} = \\ = \max \{ 11, 10 \} = 11;$$

$$t_p(5) = t_p(4) + t(4, 5) = 11 + 6 = 17.$$

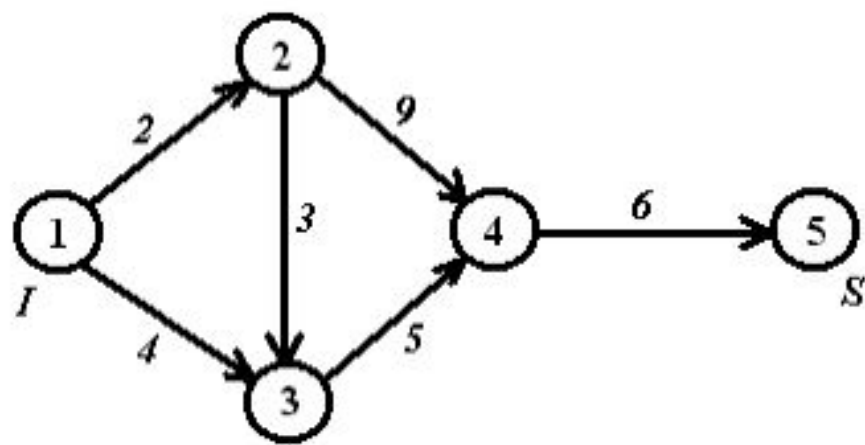
**Поздним сроком  $t_n(i)$  свершения события  $i$**  называется самый поздний момент времени, после которого остается столько времени до критического срока, сколько необходимо для завершения всех работ, следующих за этим событием:

$$t_n(i) = \min_{(i,j) \in U_i^-} (t_n(j) - t(i,j)),$$

где  $t_n(j)$  - поздний срок свершения события  $j$ ;

$t(i,j)$  - продолжительность работы  $(i,j)$ ;

$U_i^-$  - множество работ, выходящих из события  $i$ .



$$t_n(5) = 17;$$

$$t_n(4) = t_n(5) - t(4, 5) = 17 - 6 = 11;$$

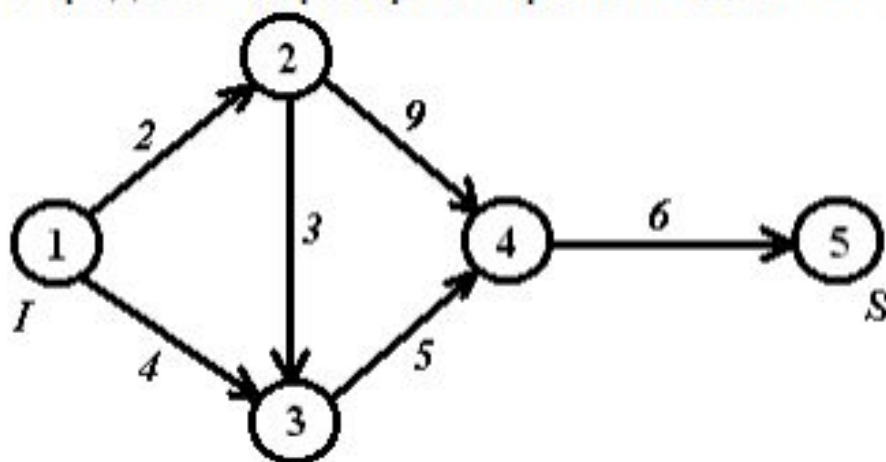
$$t_n(3) = t_n(4) - t(3, 4) = 11 - 5 = 6;$$

$$t_n(2) = \min\{(t_n(3) - t(2, 3)), (t_n(4) - t(2, 4))\} = \min\{(6 - 3), (11 - 9)\} = \\ = \min\{3, 2\} = 2;$$

$$t_n(1) = \min\{(t_n(2) - t(1, 2)), (t_n(3) - t(1, 3))\} = \min\{(2 - 2), (6 - 4)\} = \\ = \min\{0, 2\} = 0.$$

Тогда *резерв времени события i* равен:

$$R(i) = t_n(i) - t_p(i).$$



$$R(1) = t_n(1) - t_p(1) = 0 - 0 = 0 ;$$

$$R(2) = t_n(2) - t_p(2) = 2 - 2 = 0 ;$$

$$R(3) = t_n(3) - t_p(3) = 6 - 5 = 1 ;$$

$$R(4) = t_n(4) - t_p(4) = 11 - 11 = 0 ;$$

$$R(5) = t_n(5) - t_p(5) = 17 - 17 = 0 .$$