Discrete Mathematical Models

Lecture 27

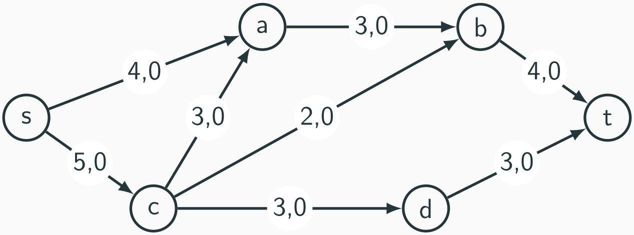
Kane Townsend

Semester 2, 2024

Maximal flow (cont)

Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1

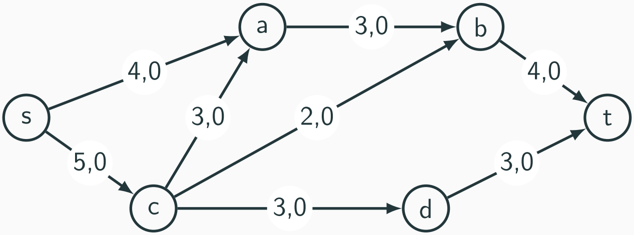
Flow F_0



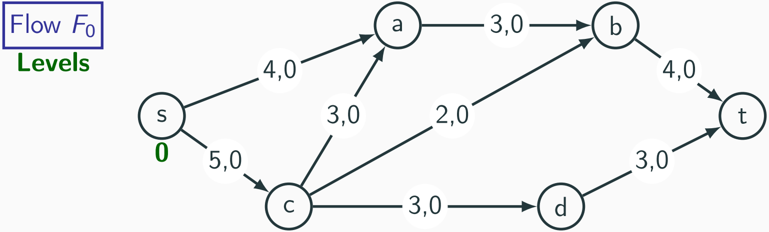
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1

Flow F_0

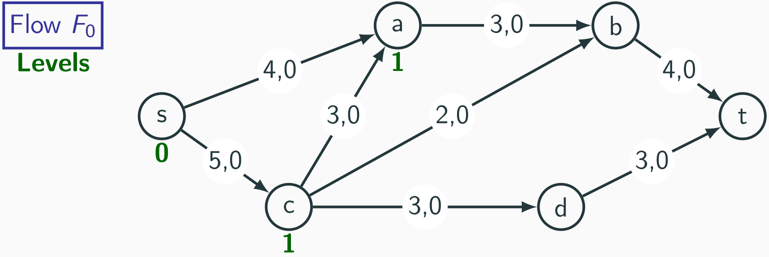
Levels



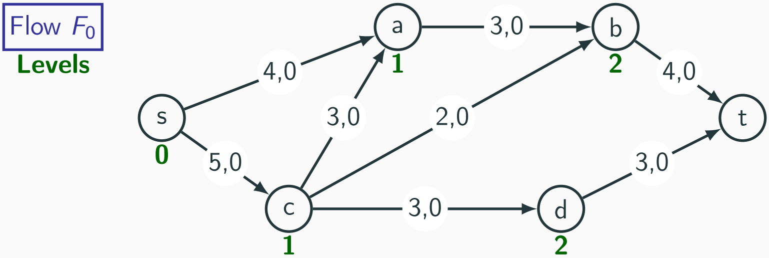
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



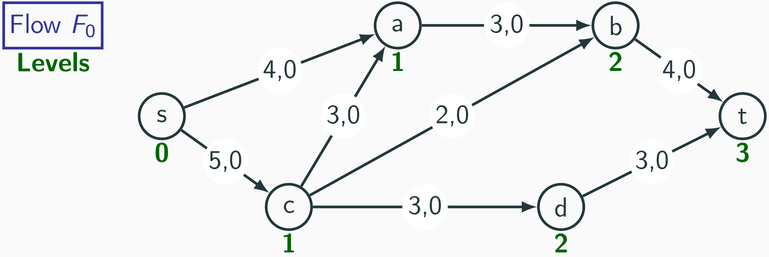
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1

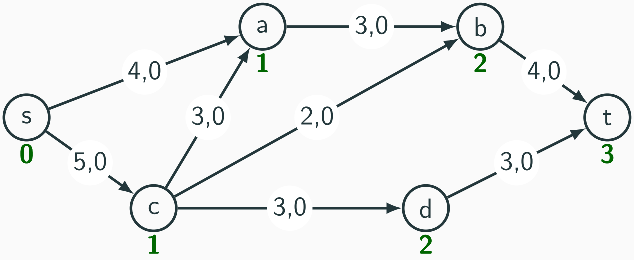


Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1

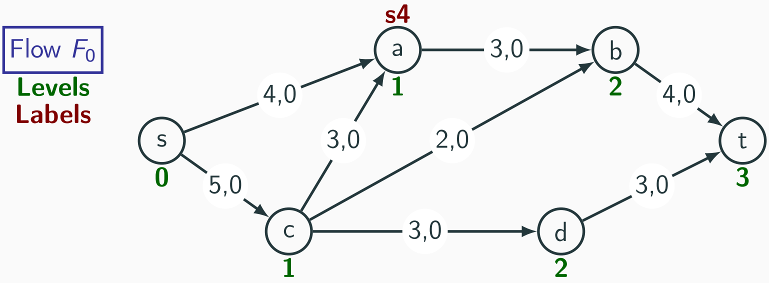
Flow F_0

Levels

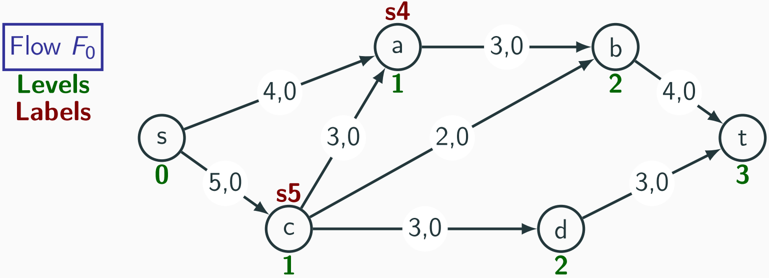
Labels



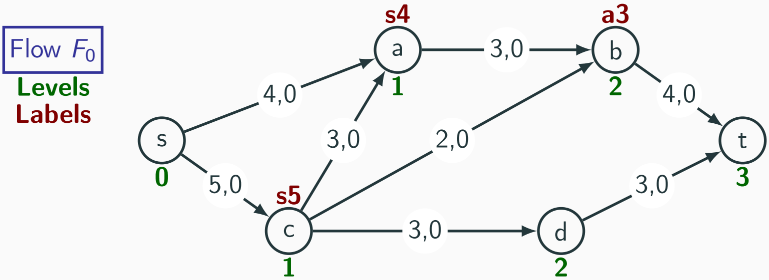
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



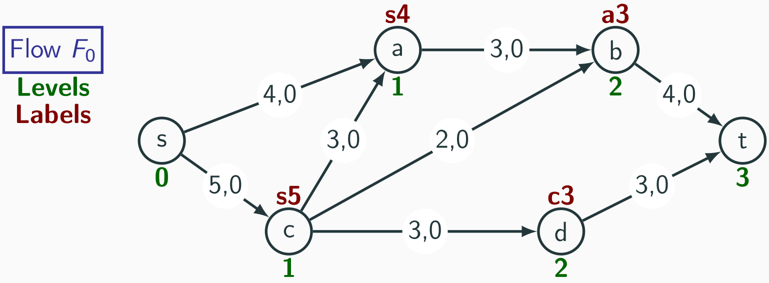
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



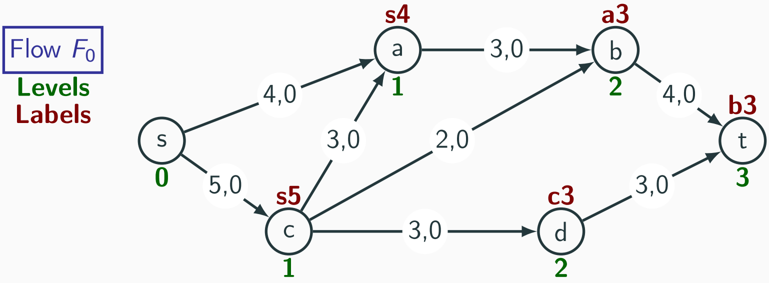
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



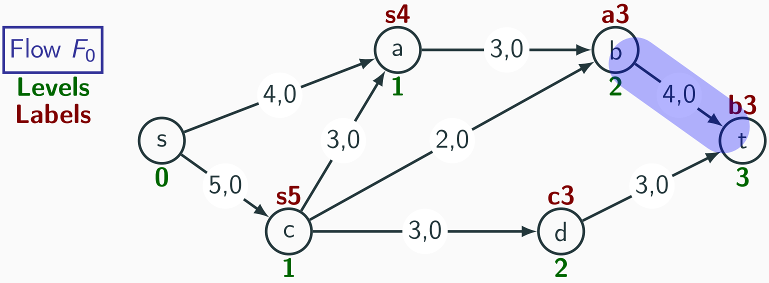
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



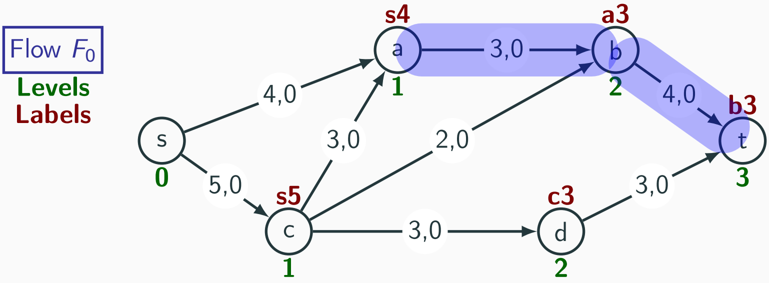
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



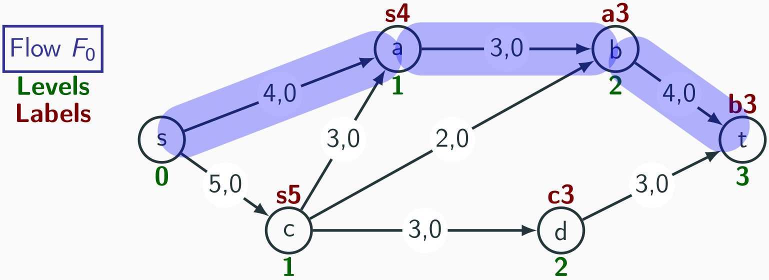
Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1



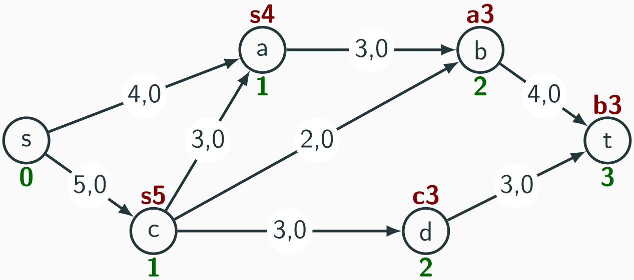
The incremental flow f_1

Vertex labelling algorithm, Example 2: Stage 1: F_0 to F_1

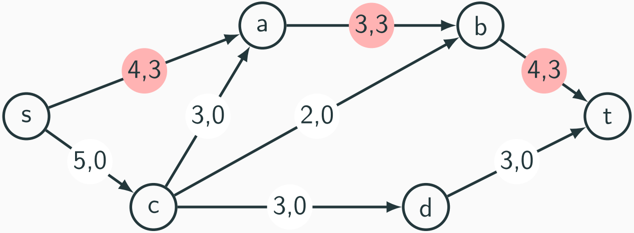
Flow F_0

Levels

Labels

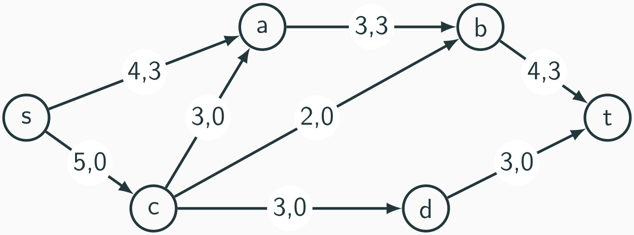


Flow F_1



Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2

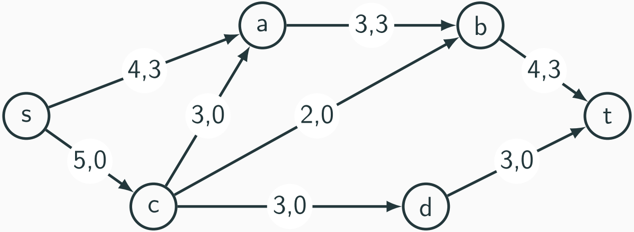
Flow F_1



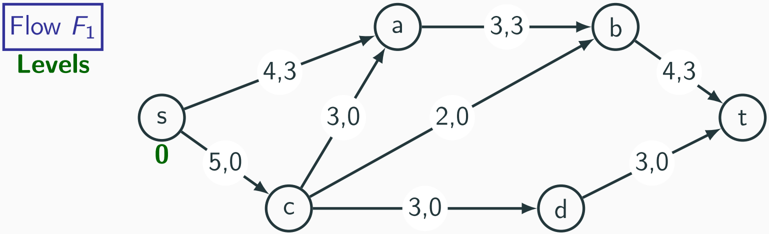
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2

Flow F_1

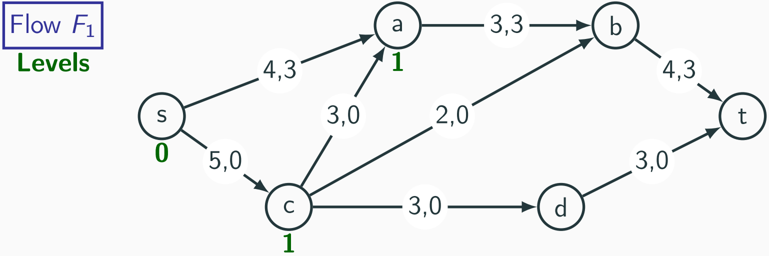
Levels



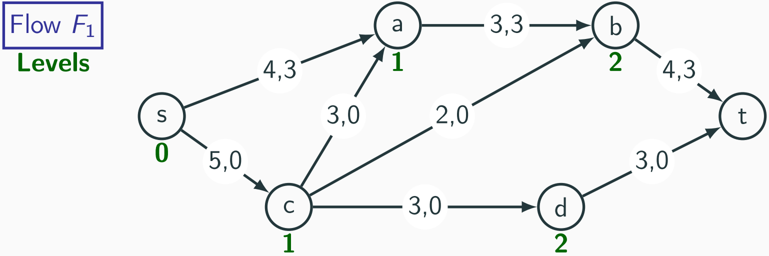
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



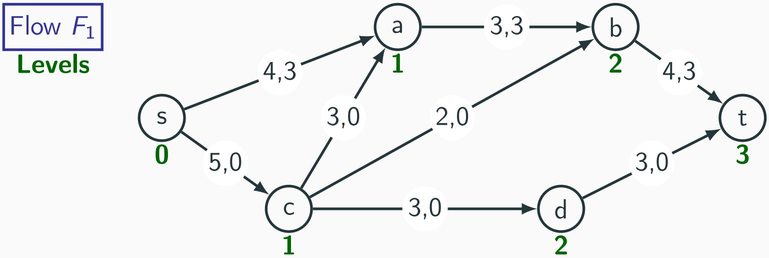
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2

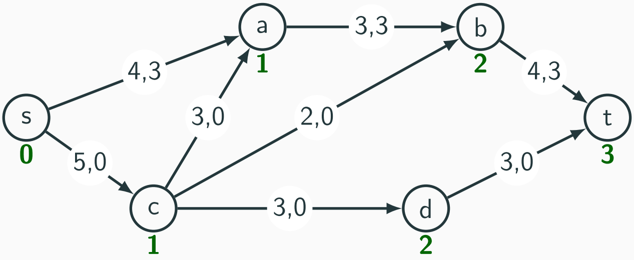


Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2

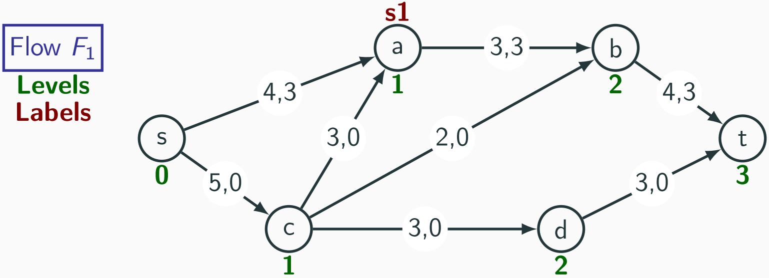
Flow F_1

Levels

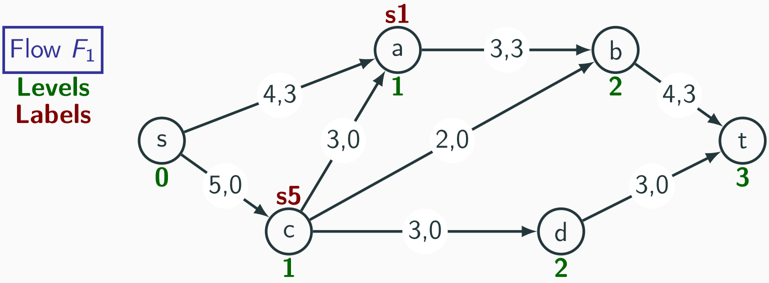
Labels



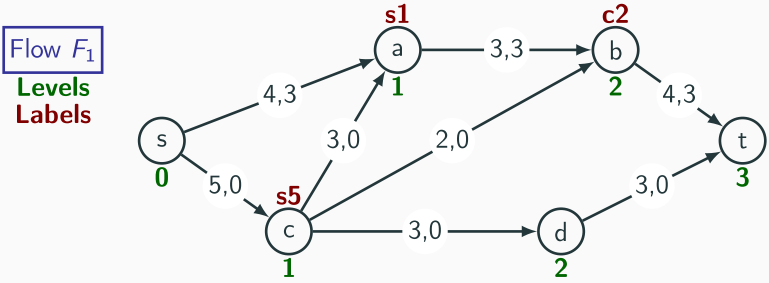
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



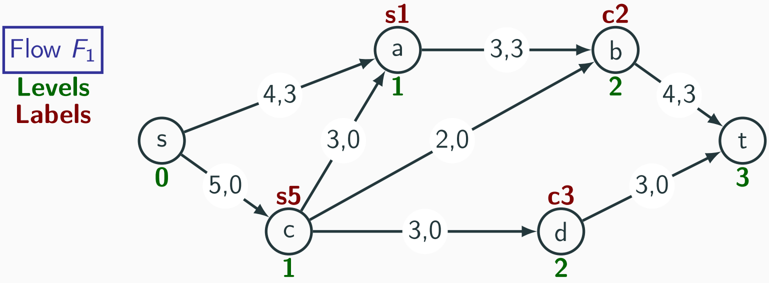
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



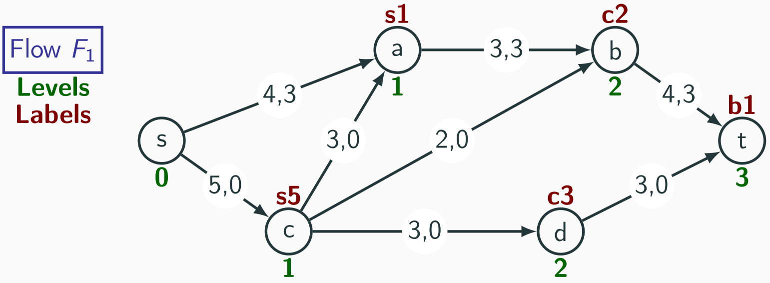
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



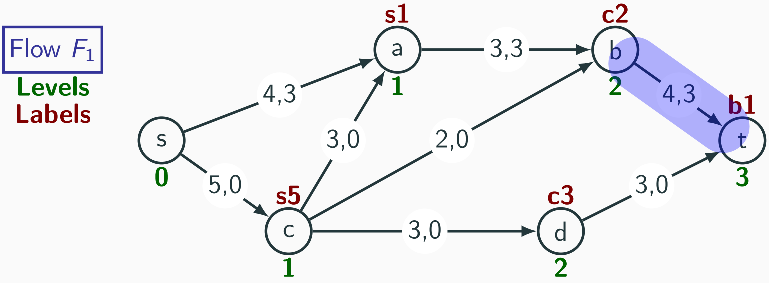
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



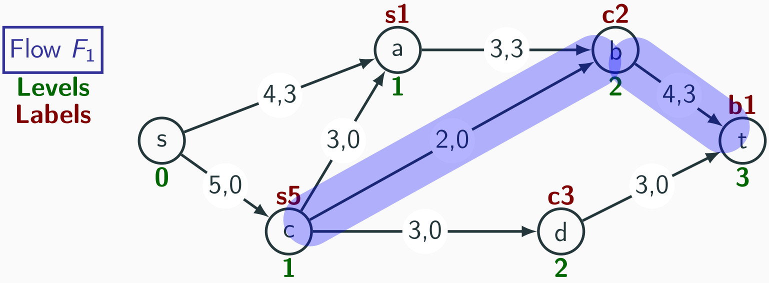
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



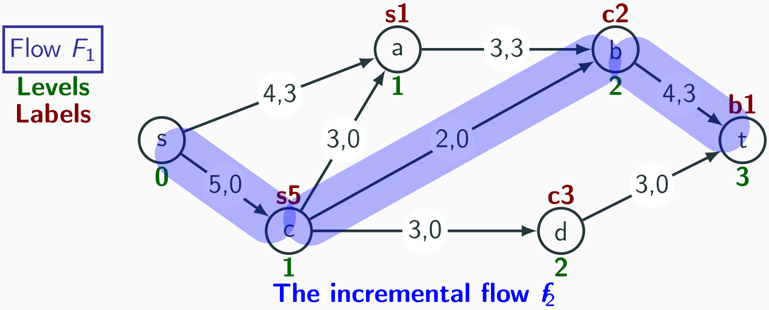
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



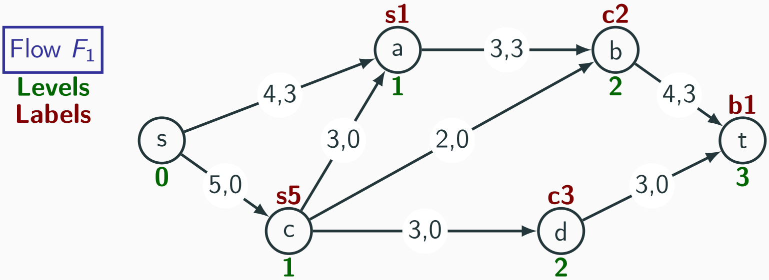
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2

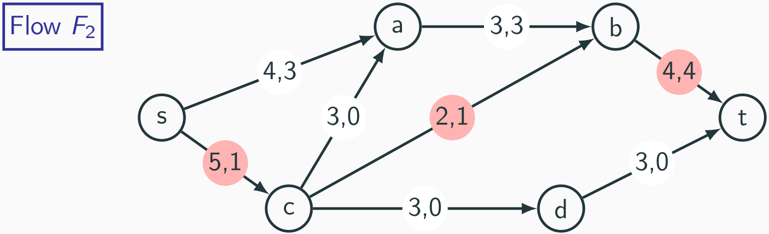


Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2



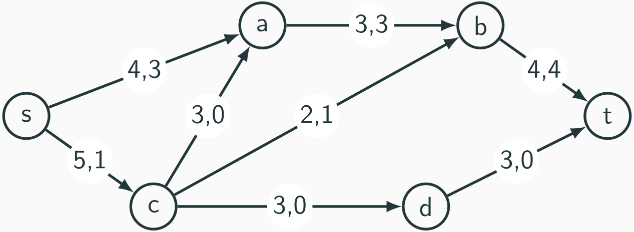
Vertex labelling algorithm, Example 2: Stage 2: F_1 to F_2





Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3

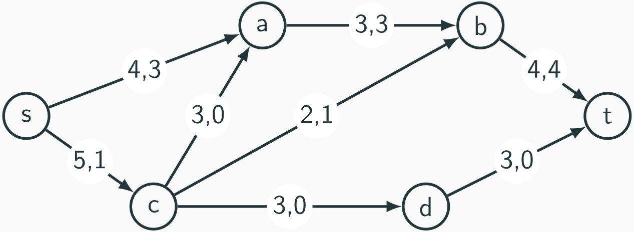
Flow F_2



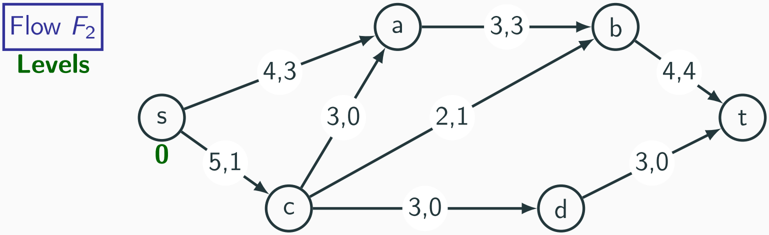
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3

Flow F_2

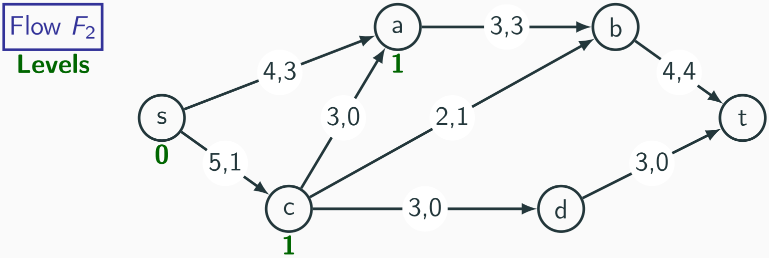
Levels



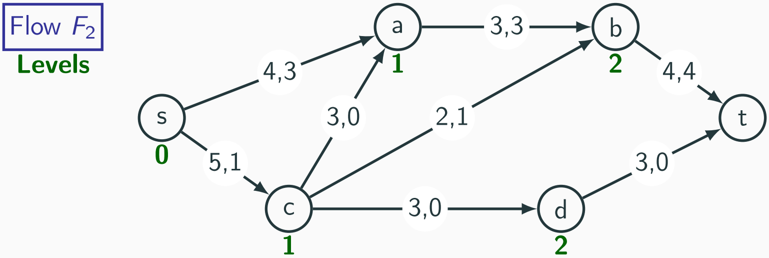
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



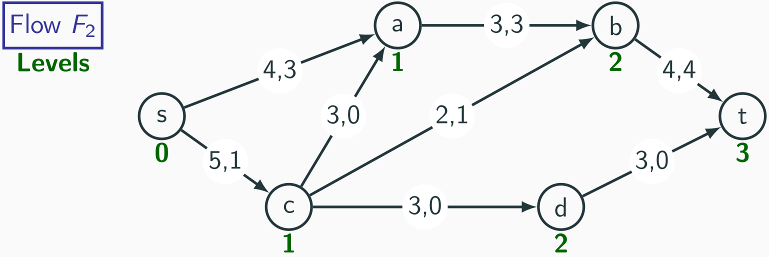
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3

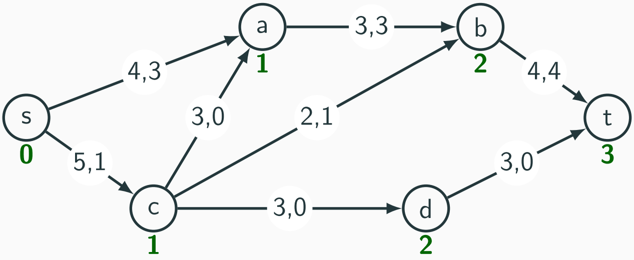


Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3

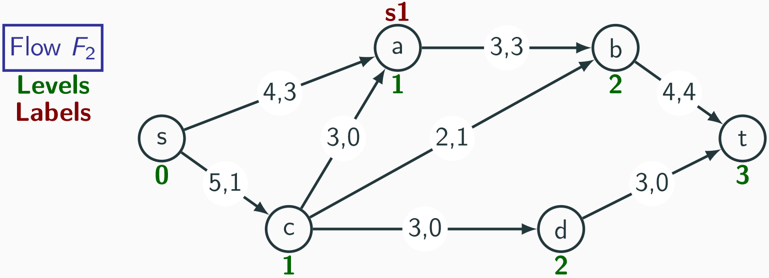
Flow F_2

Levels

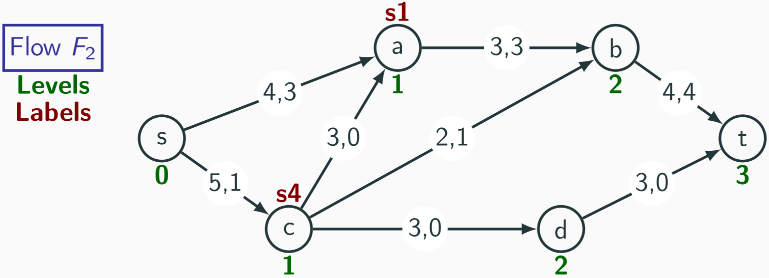
Labels



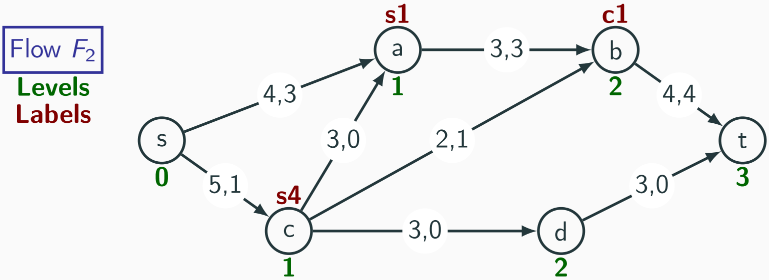
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



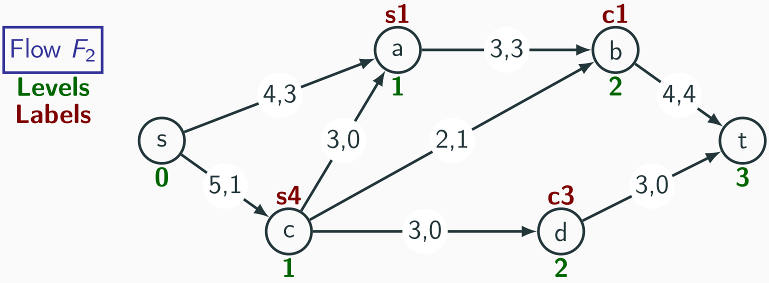
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



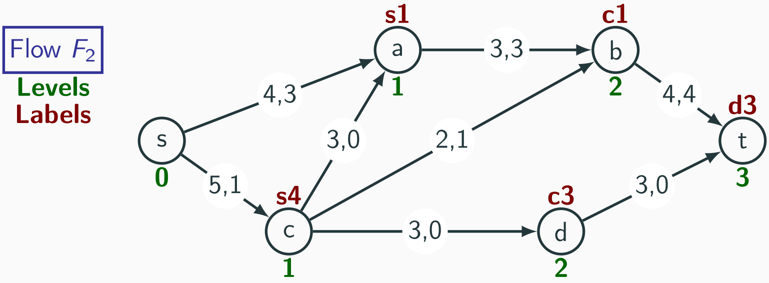
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



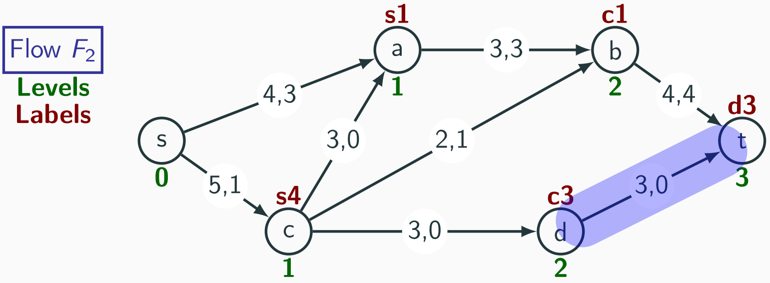
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



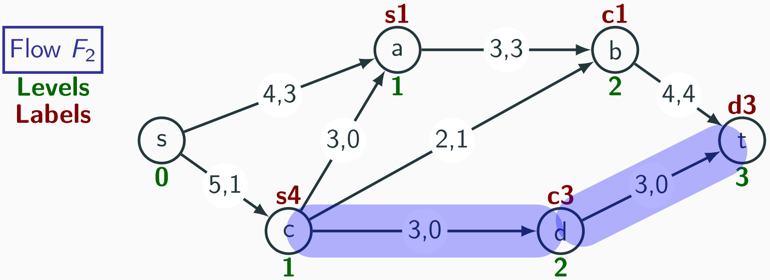
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



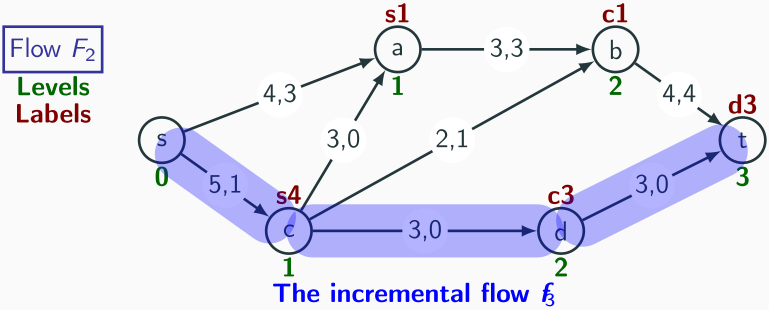
Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3



Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3

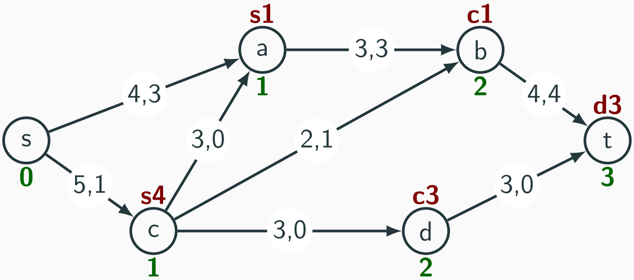


Vertex labelling algorithm, Example 2: Stage 3: F_2 to F_3

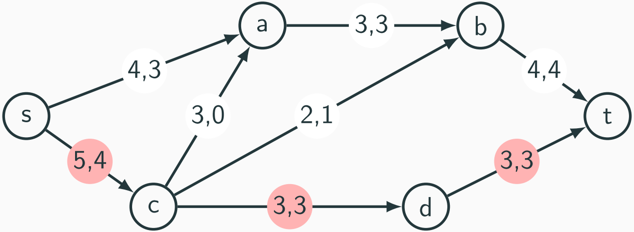
Flow F_2

Levels

Labels

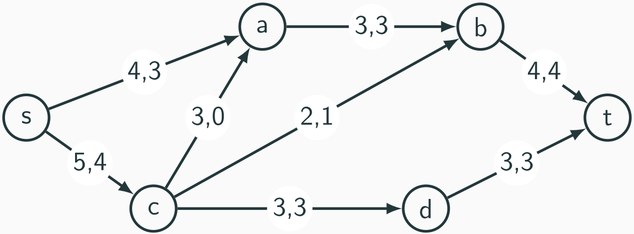


Flow F_3

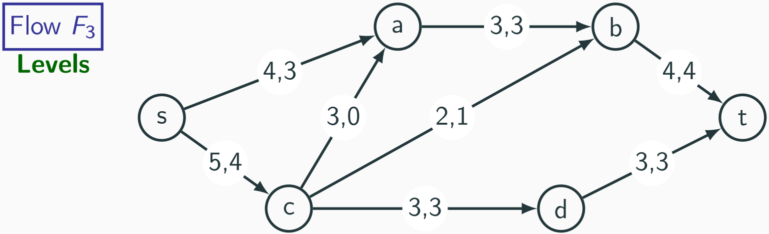


Vertex labelling algorithm, Example 2: Stage 4: F_3 is $F_{\mathbf{max}}$

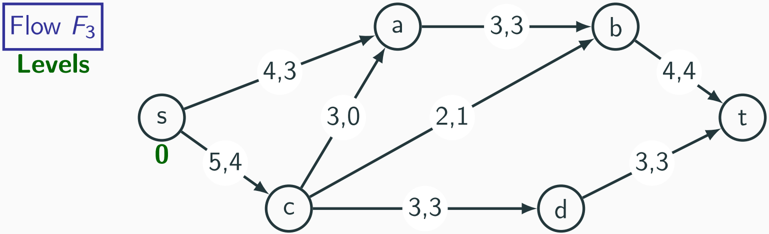
Flow F_3



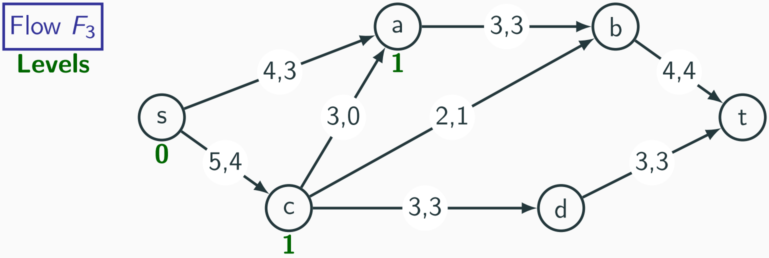
Vertex labelling algorithm, Example 2: Stage 4: F_3 is $F_{\mathbf{max}}$



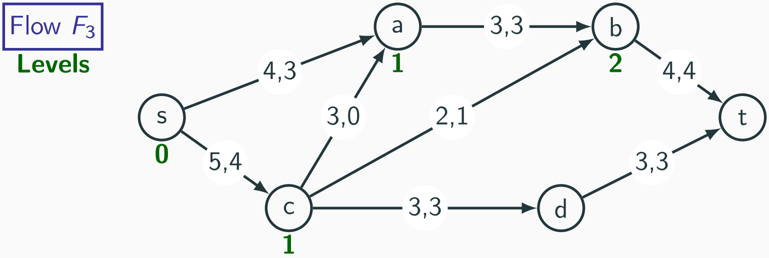
Vertex labelling algorithm, Example 2: Stage 4: F_3 is $F_{\mathbf{max}}$



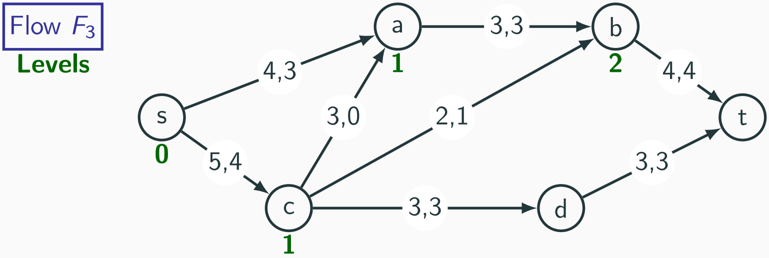
Vertex labelling algorithm, Example 2: Stage 4: F_3 is $F_{\mathbf{max}}$



Vertex labelling algorithm, Example 2: Stage 4: F_3 is $F_{\mathbf{max}}$

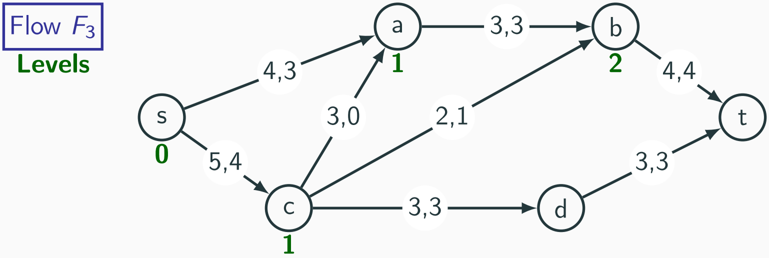


Vertex labelling algorithm, Example 2: Stage 4: F_3 is $F_{\mathbf{max}}$



No level can be assigned to t !

Vertex labelling algorithm, Example 2: Stage 4: F_3 is $F_{\mathbf{max}}$

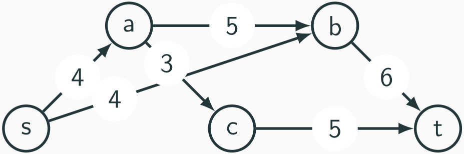


No level can be assigned to t !

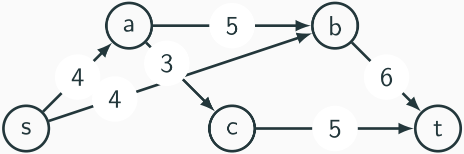
So the algorithm terminates with $F_{\max} = F_3$.

Cuts

Consider again Example 1 at right.



Cuts

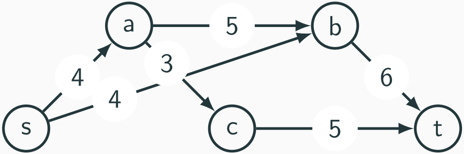


Consider again Example 1 at right.

Since the total capacity of edges leading from the source is $4 + 4 = 8$ it is clear that the maximum flow value cannot be more than 8.

Cuts

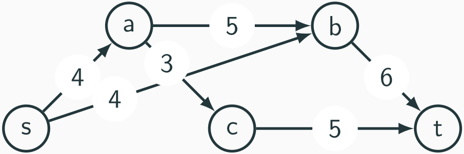
Consider again Example 1 at right.



Since the total capacity of edges leading from the source is $4 + 4 = 8$ it is clear that the maximum flow value cannot be more than 8.

In fact the maximum flow value *is* 8 as a flow with that value is easily found. (We found it using the vertex labelling algorithm, but that isn't really needed on such a simple example.)

Cuts



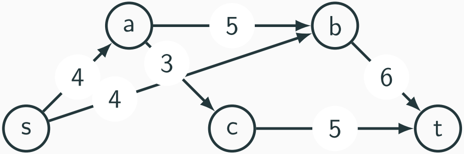
Consider again Example 1 at right.
Since the total capacity of edges leading from the source is $4 + 4 = 8$ it is clear that the maximum flow value cannot be more than 8.

In fact the maximum flow value *is* 8 as a flow with that value is easily found. (We found it using the vertex labelling algorithm, but that isn't really needed on such a simple example.)

The set of edges leading out of the source is an example of a network *cut*, so called because removing the edges would 'cut' the network in two, separating the source from the target.
The $4 + 4$ is the 'capacity' of the cut.

Cuts

Consider again Example 1 at right.



Since the total capacity of edges leading from the source is $4 + 4 = 8$ it is clear that the maximum flow value cannot be more than 8.

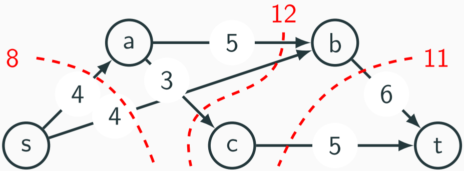
In fact the maximum flow value *is* 8 as a flow with that value is easily found. (We found it using the vertex labelling algorithm, but that isn't really needed on such a simple example.)

The set of edges leading out of the source is an example of a network *cut*, so called because removing the edges would 'cut' the network in two, separating the source from the target.
The $4 + 4$ is the 'capacity' of the cut.

A cut may be indicated by drawing a (red) line through its edges.

Cuts

Consider again Example 1 at right.



Since the total capacity of edges leading from the source is $4+4=8$ it is clear that the maximum flow value cannot be more than 8.

In fact the maximum flow value *is* 8 as a flow with that value is easily found. (We found it using the vertex labelling algorithm, but that isn't really needed on such a simple example.)

The set of edges leading out of the source is an example of a network *cut*, so called because removing the edges would 'cut' the network in two, separating the source from the target.
The $4+4$ is the 'capacity' of the cut.

A cut may be indicated by drawing a (red) line through its edges.
Some examples, with values, are shown on the digraph above.

Max Flow Min Cut

Observe that for Example 1 the minimum cut and maximum flow have the same value (8).

Max Flow Min Cut

Observe that for Example 1 the minimum cut and maximum flow have the same value (8). As you probably expect, this is in fact true for all transport networks, as given by the theorem below.

Observe that for Example 1 the minimum cut and maximum flow have the same value (8). As you probably expect, this is in fact true for all transport networks, as given by the theorem below.

First a formal definition of ‘cut’:

Max Flow Min Cut

Observe that for Example 1 the minimum cut and maximum flow have the same value (8). As you probably expect, this is in fact true for all transport networks, as given by the theorem below.

First a formal definition of 'cut':
A set K of edges in a transport network D is called a **cut** when there is a partition $\{S, T\}$ of $V(D)$ with $s \in S$ and $t \in T$ such that K comprises all edges of D that start in S and finish in T; *i.e.* $K = E(D) \cap (S \times T)$.

Max Flow Min Cut

Observe that for Example 1 the minimum cut and maximum flow have the same value (8). As you probably expect, this is in fact true for all transport networks, as given by the theorem below.

First a formal definition of 'cut':
A set K of edges in a transport network D is called a **cut** when there is a partition $\{S, T\}$ of $V(D)$ with $s \in S$ and $t \in T$ such that K comprises all edges of D that start in S and finish in T; *i.e.* $K = E(D) \cap (S \times T)$.

The **capacity of a cut** K is $\sum_{e \in K} C(e)$.

Max Flow Min Cut

Observe that for Example 1 the minimum cut and maximum flow have the same value (8). As you probably expect, this is in fact true for all transport networks, as given by the theorem below.

First a formal definition of 'cut':
A set K of edges in a transport network D is called a **cut** when there is a partition $\{S, T\}$ of $V(D)$ with $s \in S$ and $t \in T$ such that K comprises all edges of D that start in S and finish in T; *i.e.* $K = E(D) \cap (S \times T)$.

The **capacity of a cut** K is $\sum_{e \in K} C(e)$.

Max flow min cut theorem: For any transport network:

$$\boxed{\textbf{maximum flow value} \quad = \quad \textbf{minimum cut capacity}}$$

Max Flow Min Cut

Observe that for Example 1 the minimum cut and maximum flow have the same value (8). As you probably expect, this is in fact true for all transport networks, as given by the theorem below.

First a formal definition of 'cut':
A set K of edges in a transport network D is called a **cut** when there is a partition $\{S, T\}$ of $V(D)$ with $s \in S$ and $t \in T$ such that K comprises all edges of D that start in S and finish in T; *i.e.* $K = E(D) \cap (S \times T)$.

The **capacity of a cut** K is $\sum_{e \in K} C(e)$.

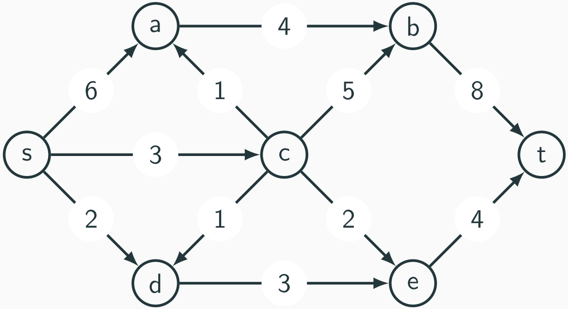
Max flow min cut theorem: For any transport network:

$$\boxed{\textbf{maximum flow value} \quad = \quad \textbf{minimum cut capacity}}$$

Though highly plausible, this theorem is little tricky to prove, and the proof will be omitted, as will the proof that the vertex labelling algorithm always finds a maximum flow.

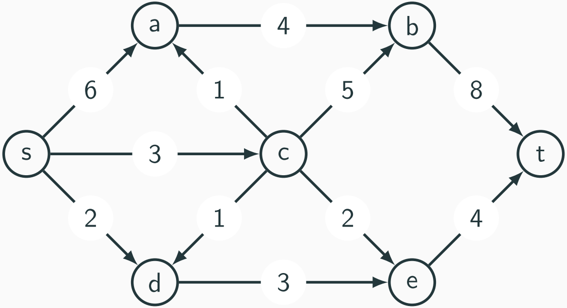
Max flow min cut: Class example 1

What is the maximum flow value for this transport network?



Max flow min cut: Class example 1

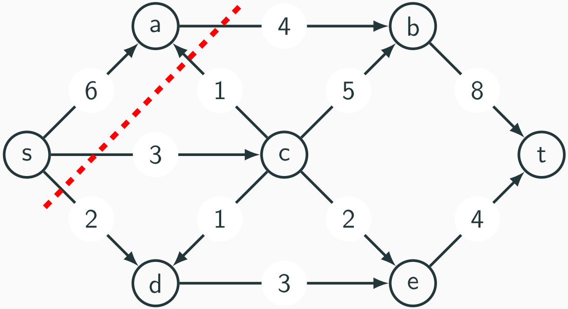
What is the maximum flow value for this transport network?



Answer:

Max flow min cut: Class example 1

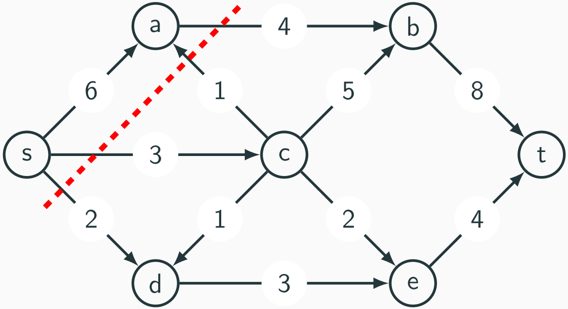
What is the maximum flow value for this transport network?



Answer: After some searching we find the minimum cut shown, for which $S = \{s, a\}$ and $T = \{b, c, d, e, t\}$. This is given by cut $K = \{(s, d), (s, c), (a, b)\}$

Max flow min cut: Class example 1

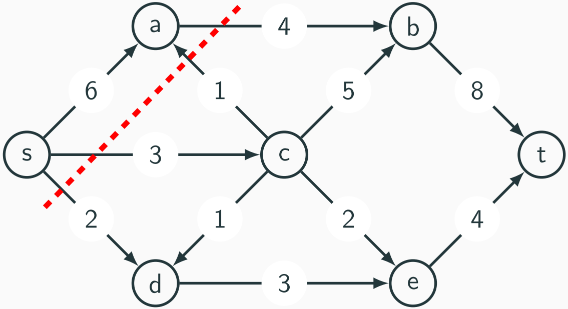
What is the maximum flow value for this transport network?



Answer: After some searching we find the minimum cut shown, for which $S = \{s, a\}$ and $T = \{b, c, d, e, t\}$. This is given by cut
$K = \{(s, d), (s, c), (a, b)\}$
The capacity of this cut is 4+3+2=9, so this is the maximum flow.

Max flow min cut: Class example 1

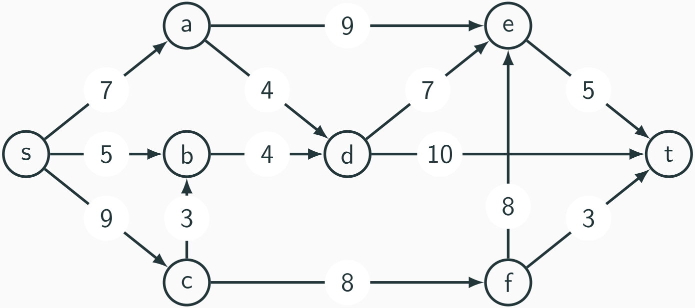
What is the maximum flow value for this transport network?



Answer: After some searching we find the minimum cut shown, for which $S = \{s, a\}$ and $T = \{b, c, d, e, t\}$. This is given by cut
$K = \{(s, d), (s, c), (a, b)\}$
The capacity of this cut is 4+3+2=9, so this is the maximum flow.
Note: Make sure $K = E(D) \cap (S \times T)$

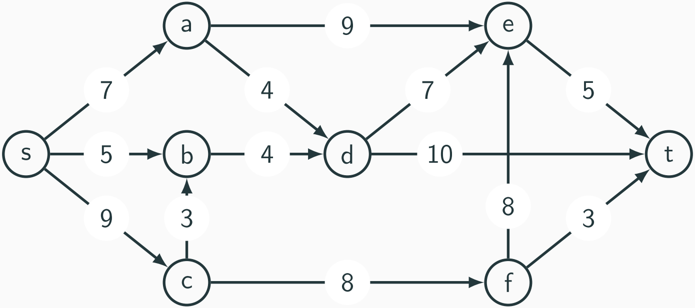
Max flow min cut: Class example 2

What is the maximum flow value for this transport network?



Max flow min cut: Class example 2

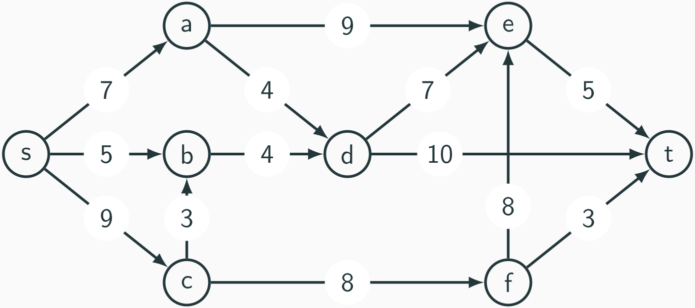
What is the maximum flow value for this transport network?



Hint:

Max flow min cut: Class example 2

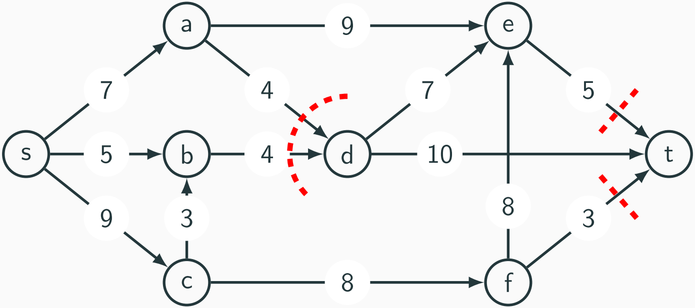
What is the maximum flow value for this transport network?



Hint: This time the minimum cut cannot be drawn as a single line!

Max flow min cut: Class example 2

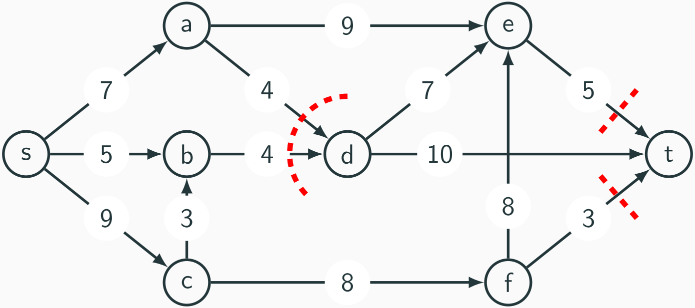
What is the maximum flow value for this transport network?



Hint: This time the minimum cut cannot be drawn as a single line!
Answer: Use all the edges from $S = \{s, a, b, c, e, f\}$ to $T = \{d, t\}$. This is given by cut $K = \{(a, d), (b, d), (f, t), (e, t)\}$

Max flow min cut: Class example 2

What is the maximum flow value for this transport network?



Hint: This time the minimum cut cannot be drawn as a single line!
Answer: Use all the edges from $S = \{s, a, b, c, e, f\}$ to $T = \{d, t\}$. This is given by cut $K = \{(a, d), (b, d), (f, t), (e, t)\}$
The capacity of this cut is 4+4+5+3=16, so this is the max flow.

Virtual flows

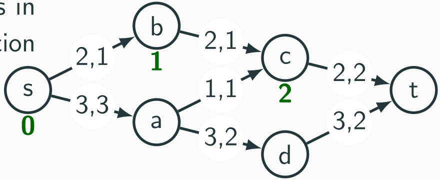
Introduction to virtual flows

The next example (Example 3) will show that in order to always achieve a maximum flow the labelling algorithm needs a modification.

Introduction to virtual flows

The next example (Example 3) will show that in order to always achieve a maximum flow the labelling algorithm needs a modification.

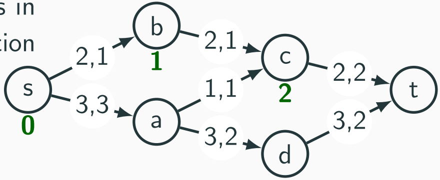
Part way through assigning levels in Stage 4 (of Example 3) the situation becomes as shown at right.



Introduction to virtual flows

The next example (Example 3) will show that in order to always achieve a maximum flow the labelling algorithm needs a modification.

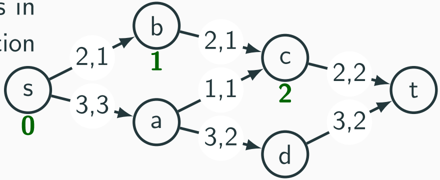
Part way through assigning levels in Stage 4 (of Example 3) the situation becomes as shown at right. We are 'stuck' at vertex c, but so far we only have a flow value of 4, whereas the min cut value is 5.



Introduction to virtual flows

The next example (Example 3) will show that in order to always achieve a maximum flow the labelling algorithm needs a modification.

Part way through assigning levels in Stage 4 (of Example 3) the situation becomes as shown at right. We are 'stuck' at vertex c, but so far we only have a flow value of 4, whereas the min cut value is 5.

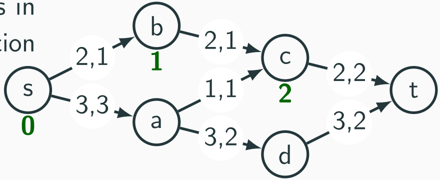


To escape from this we need to 'divert' the flow a→c→t to a→d→t, thereby allowing another 1 unit of flow s→b→c→t.

Introduction to virtual flows

The next example (Example 3) will show that in order to always achieve a maximum flow the labelling algorithm needs a modification.

Part way through assigning levels in Stage 4 (of Example 3) the situation becomes as shown at right. We are 'stuck' at vertex c, but so far we only have a flow value of 4, whereas the min cut value is 5.



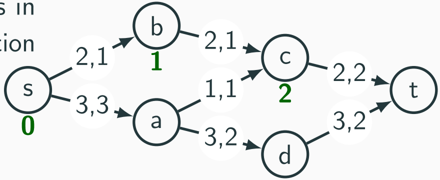
To escape from this we need to 'divert' the flow a$\rightarrow$c$\rightarrow$t to a$\rightarrow$d$\rightarrow$t, thereby allowing another 1 unit of flow s$\rightarrow$b$\rightarrow$c$\rightarrow$t.

The algorithm accomplishes this diversion by allowing a 'virtual flow' of 1 unit c$\rightarrow$a so that, in particular, vertex a becomes level 3.

Introduction to virtual flows

The next example (Example 3) will show that in order to always achieve a maximum flow the labelling algorithm needs a modification.

Part way through assigning levels in Stage 4 (of Example 3) the situation becomes as shown at right. We are 'stuck' at vertex c, but so far we only have a flow value of 4, whereas the min cut value is 5.



To escape from this we need to 'divert' the flow a→c→t to a→d→t, thereby allowing another 1 unit of flow s→b→c→t.

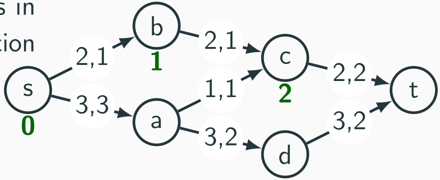
The algorithm accomplishes this diversion by allowing a 'virtual flow' of 1 unit c→a so that, in particular, vertex a becomes level 3.

What happens after that is shown as Stage 4 unfolds in the full description of the use of the algorithm in Example 3 that follows.

Introduction to virtual flows

The next example (Example 3) will show that in order to always achieve a maximum flow the labelling algorithm needs a modification.

Part way through assigning levels in Stage 4 (of Example 3) the situation becomes as shown at right. We are 'stuck' at vertex c, but so far we only have a flow value of 4, whereas the min cut value is 5.



To escape from this we need to 'divert' the flow a→c→t to a→d→t, thereby allowing another 1 unit of flow s→b→c→t.

The algorithm accomplishes this diversion by allowing a 'virtual flow' of 1 unit c→a so that, in particular, vertex a becomes level 3.

What happens after that is shown as Stage 4 unfolds in the full description of the use of the algorithm in Example 3 that follows.

First, the definition and an explanation of how the algorithm is modified.

The complete vertex labelling algorithm

Let (u,v) be a (directed) edge in a transport network D, and suppose there is currently a flow of $f > 0$ along this edge. The vertex labelling algorithm can reduce this flow to $g < f$ by introducing a **virtual flow** of $f - g$ in the opposite direction, *i.e.* from v to u.

The complete vertex labelling algorithm

Let (u,v) be a (directed) edge in a transport network D, and suppose there is currently a flow of $f > 0$ along this edge. The vertex labelling algorithm can reduce this flow to $g < f$ by introducing a **virtual flow** of $f - g$ in the opposite direction, *i.e.* from v to u.

Virtual flows are *always* in a direction counter to that indicated on the digraph D, and if part of an incremental flow will *reduce* the flow in the relevant leg of the path.

The complete vertex labelling algorithm

Let (u,v) be a (directed) edge in a transport network D, and suppose there is currently a flow of $f > 0$ along this edge. The vertex labelling algorithm can reduce this flow to $g < f$ by introducing a **virtual flow** of $f - g$ in the opposite direction, *i.e.* from v to u.

Virtual flows are *always* in a direction counter to that indicated on the digraph D, and if part of an incremental flow will *reduce* the flow in the relevant leg of the path.

The modification to the algorithm, affecting the assignment of levels and labels, is an amendment to the definition of spare capacity.

The complete vertex labelling algorithm

Let (u,v) be a (directed) edge in a transport network D, and suppose there is currently a flow of $f > 0$ along this edge. The vertex labelling algorithm can reduce this flow to $g < f$ by introducing a **virtual flow** of $f - g$ in the opposite direction, *i.e.* from v to u.

Virtual flows are *always* in a direction counter to that indicated on the digraph D, and if part of an incremental flow will *reduce* the flow in the relevant leg of the path.

The modification to the algorithm, affecting the assignment of levels and labels, is an amendment to the definition of spare capacity.

For vertices u,v of D, where D has capacity and flow functions C, F:

$$\boxed{S((\mathrm{u,v})) = \begin{cases} C((\mathrm{u,v})) - F((\mathrm{u,v})) & \text{if } (\mathrm{u,v}) \in E(D) \\ F((\mathrm{v,u})) & \text{if } (\mathrm{v,u}) \in E(D) \\ 0 & \text{otherwise} \end{cases}}$$

The complete vertex labelling algorithm

Let (u,v) be a (directed) edge in a transport network D, and suppose there is currently a flow of $f > 0$ along this edge. The vertex labelling algorithm can reduce this flow to $g < f$ by introducing a **virtual flow** of $f - g$ in the opposite direction, *i.e.* from v to u.

Virtual flows are *always* in a direction counter to that indicated on the digraph D, and if part of an incremental flow will *reduce* the flow in the relevant leg of the path.

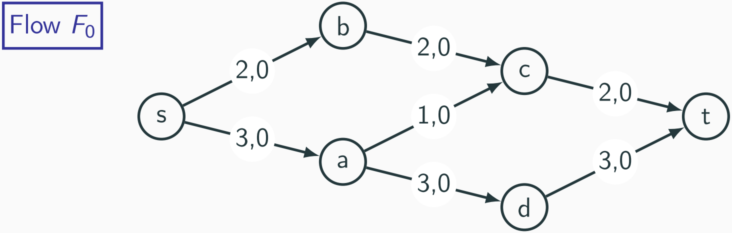
The modification to the algorithm, affecting the assignment of levels and labels, is an amendment to the definition of spare capacity.

For vertices u,v of D, where D has capacity and flow functions C, F:

$$\boxed{S((\mathrm{u,v})) = \begin{cases} C((\mathrm{u,v})) - F((\mathrm{u,v})) & \text{if } (\mathrm{u,v}) \in E(D) \\ F((\mathrm{v,u})) & \text{if } (\mathrm{v,u}) \in E(D) \\ 0 & \text{otherwise} \end{cases}}$$

When $(\mathrm{v,u}) \in E(D)$, $S((\mathrm{u,v}))$ is called a **virtual capacity**.

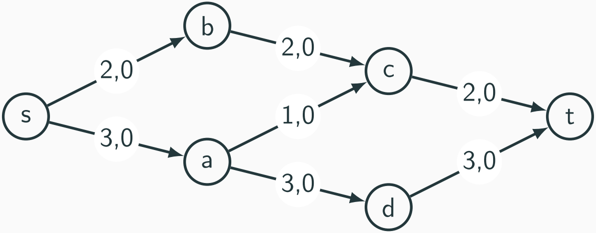
Vertex labelling algorithm, Example 3



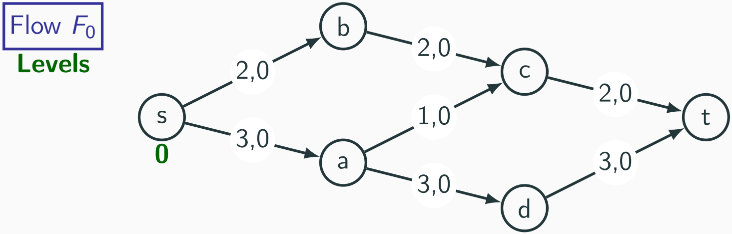
Vertex labelling algorithm, Example 3

$\boxed{\text{Flow } F_0}$

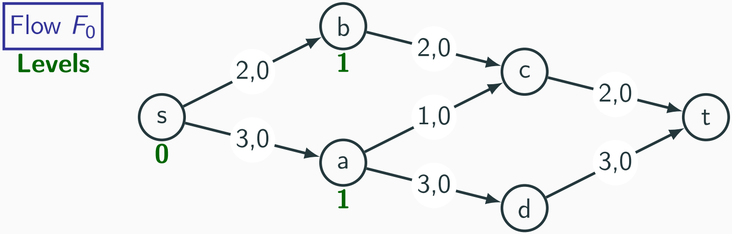
Levels



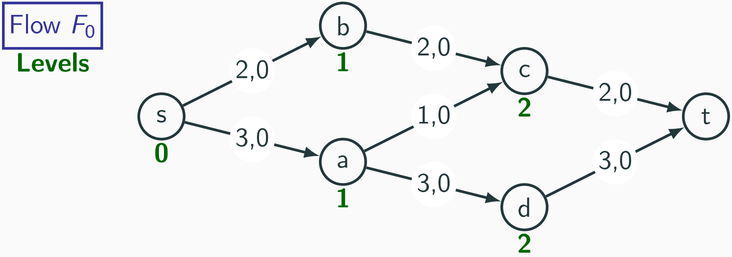
Vertex labelling algorithm, Example 3



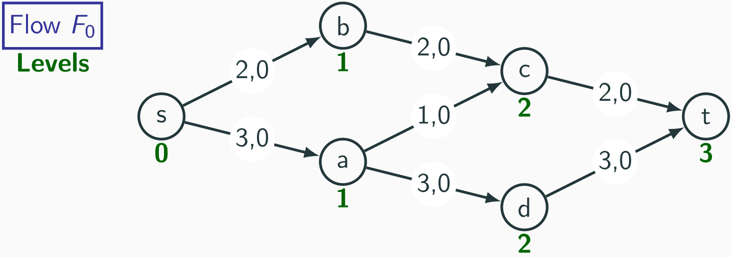
Vertex labelling algorithm, Example 3



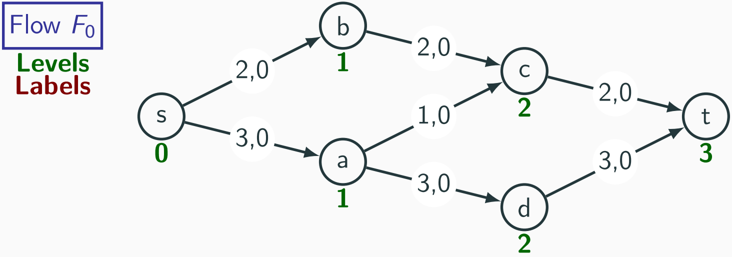
Vertex labelling algorithm, Example 3



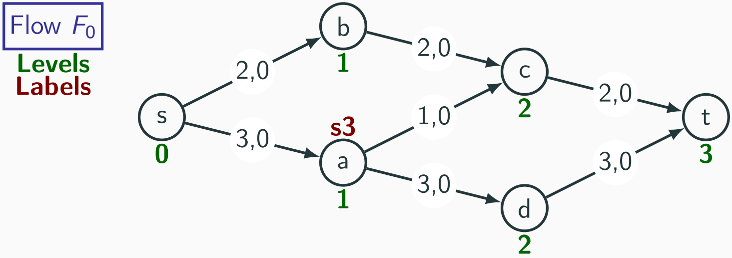
Vertex labelling algorithm, Example 3



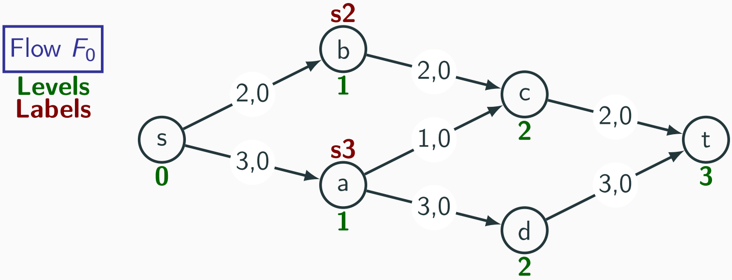
Vertex labelling algorithm, Example 3



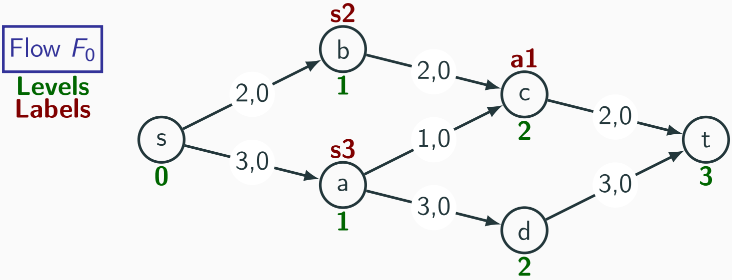
Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3



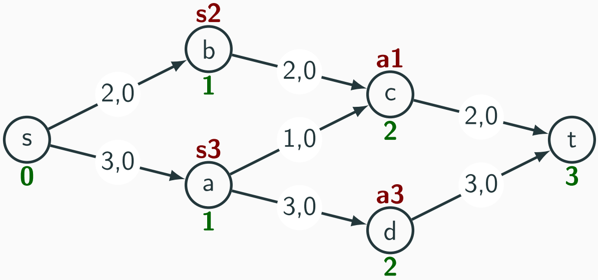
Vertex labelling algorithm, Example 3



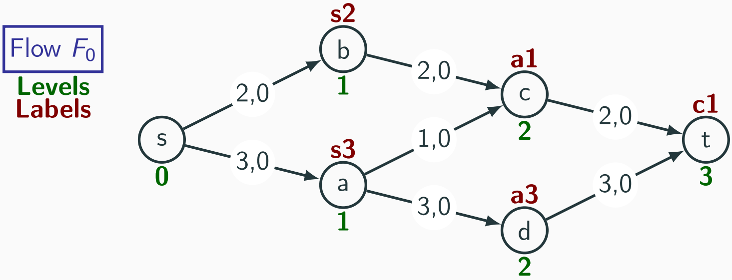
Vertex labelling algorithm, Example 3

Flow F_0

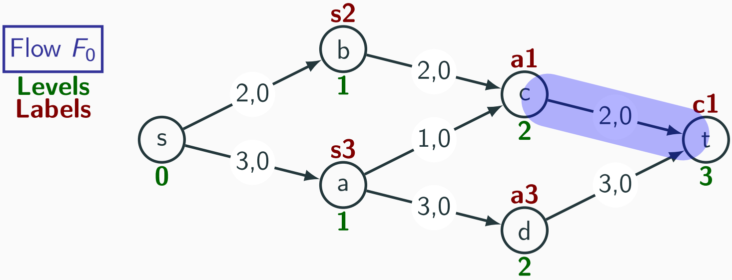
Levels
Labels



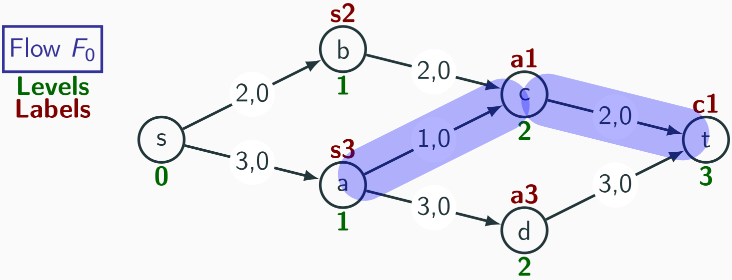
Vertex labelling algorithm, Example 3



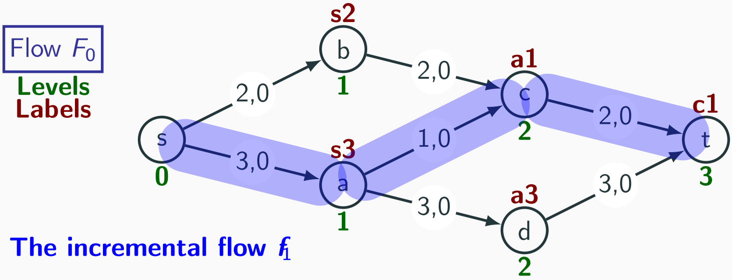
Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3

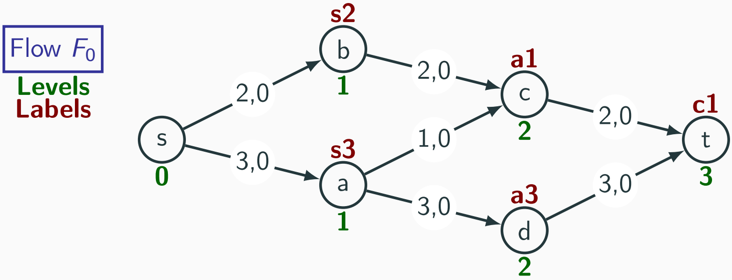


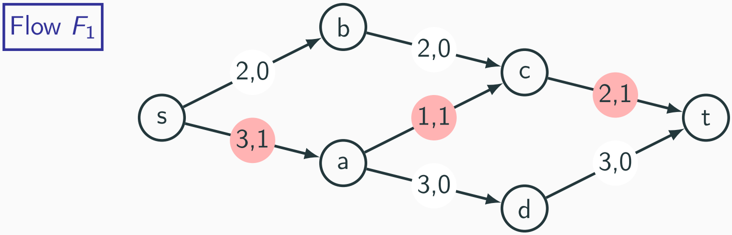
Vertex labelling algorithm, Example 3



The incremental flow f_1

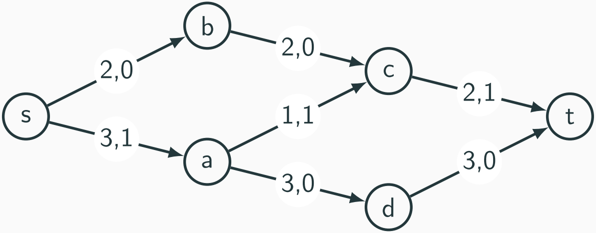
Vertex labelling algorithm, Example 3



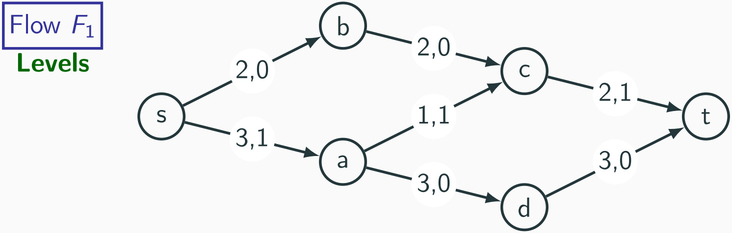


Vertex labelling algorithm, Example 3

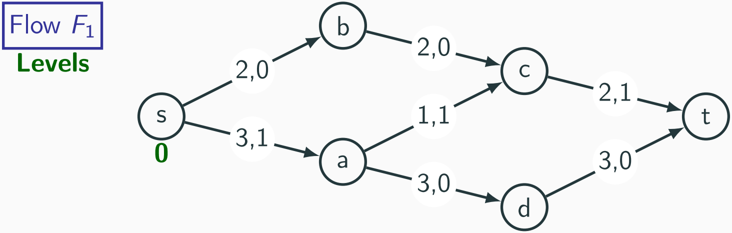
Flow F_1



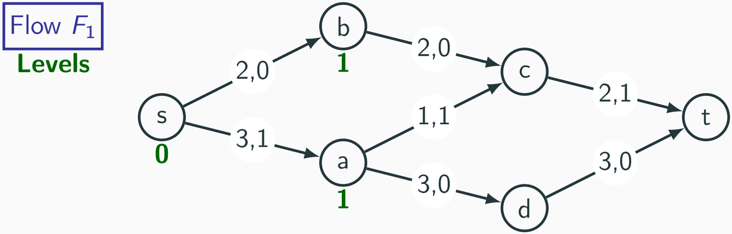
Vertex labelling algorithm, Example 3



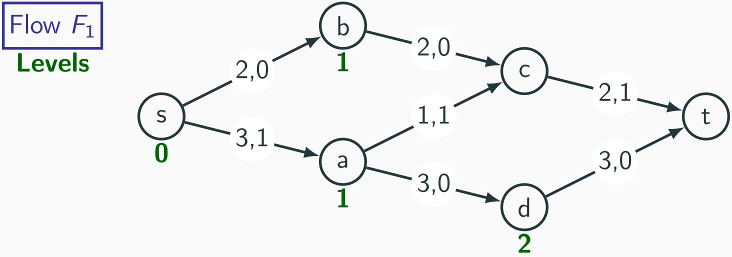
Vertex labelling algorithm, Example 3



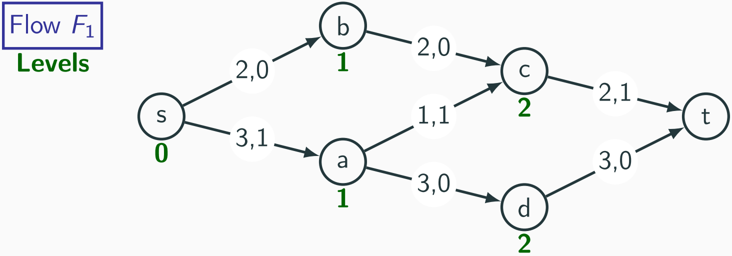
Vertex labelling algorithm, Example 3



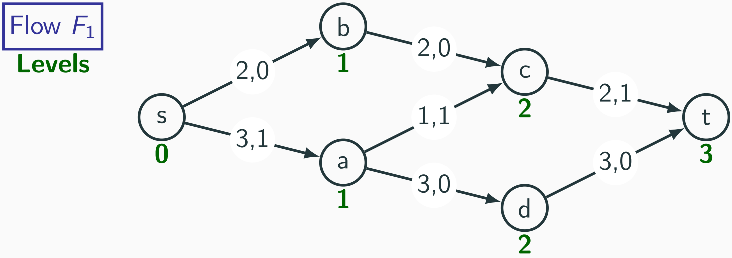
Vertex labelling algorithm, Example 3



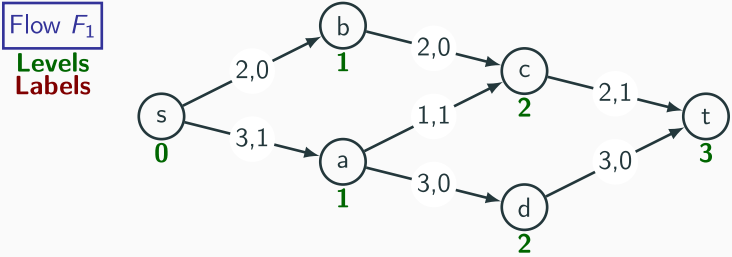
Vertex labelling algorithm, Example 3



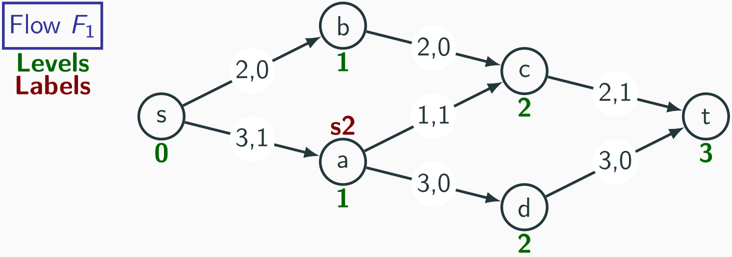
Vertex labelling algorithm, Example 3



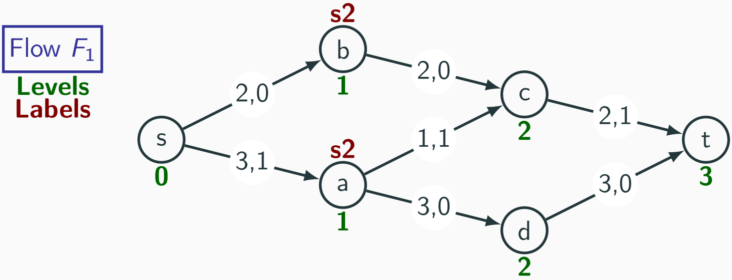
Vertex labelling algorithm, Example 3



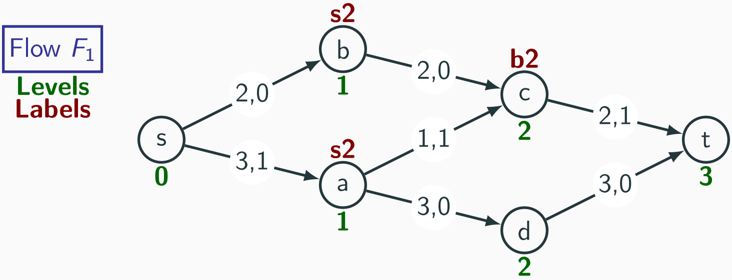
Vertex labelling algorithm, Example 3



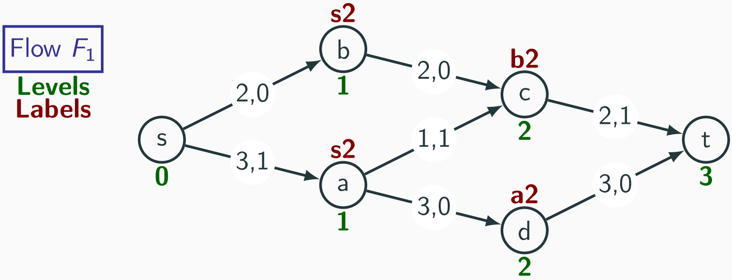
Vertex labelling algorithm, Example 3



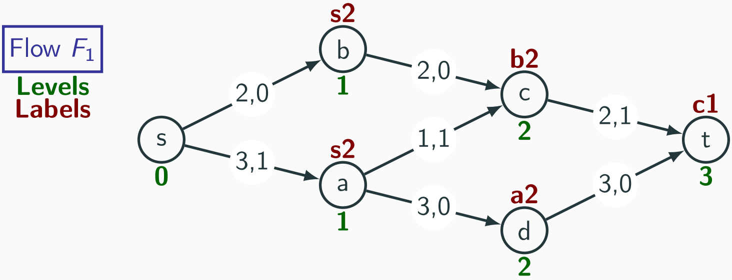
Vertex labelling algorithm, Example 3



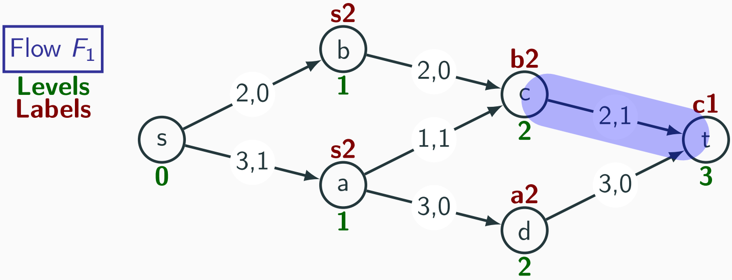
Vertex labelling algorithm, Example 3



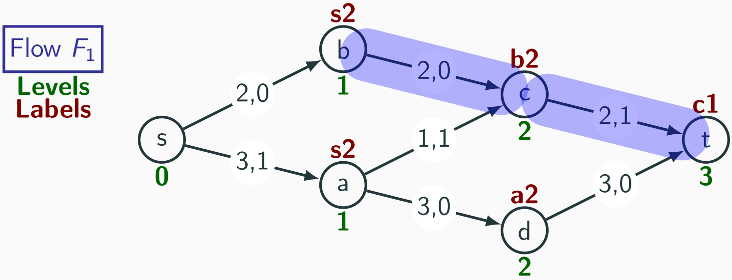
Vertex labelling algorithm, Example 3



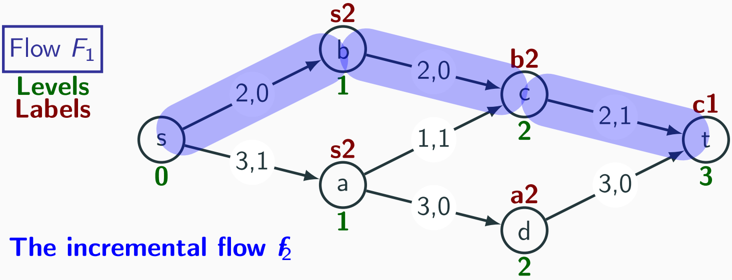
Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3

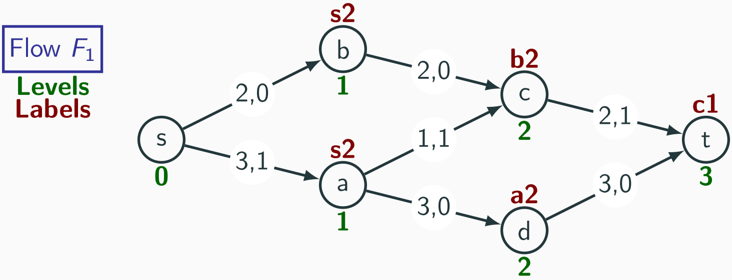


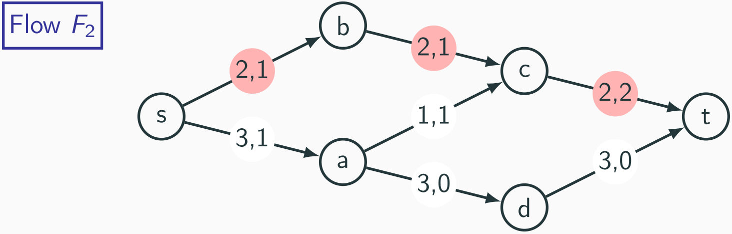
Vertex labelling algorithm, Example 3



The incremental flow f_2

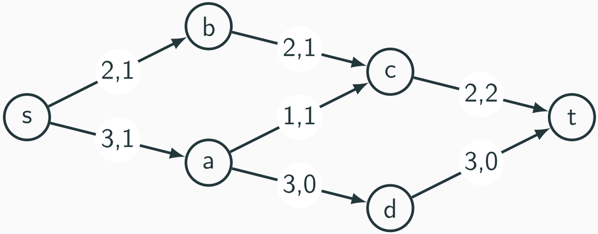
Vertex labelling algorithm, Example 3



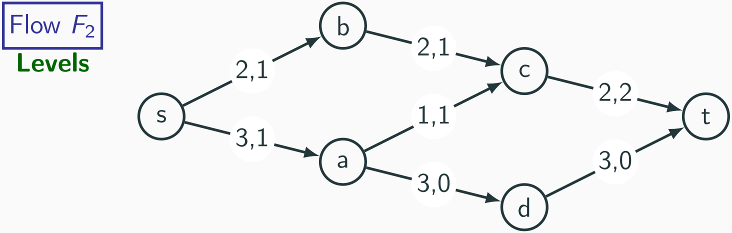


Vertex labelling algorithm, Example 3

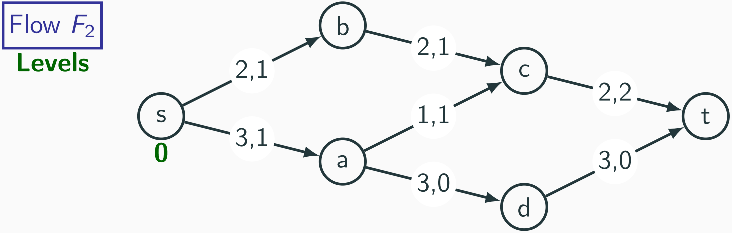
Flow F_2



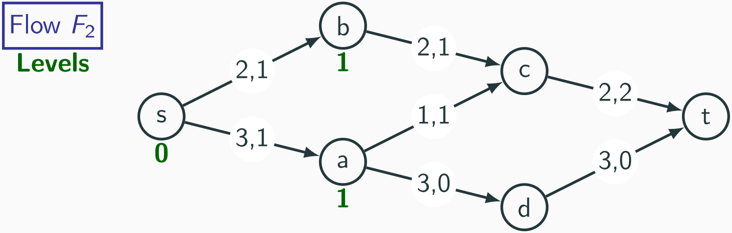
Vertex labelling algorithm, Example 3



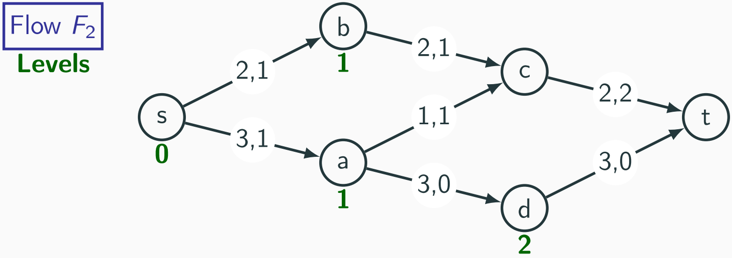
Vertex labelling algorithm, Example 3



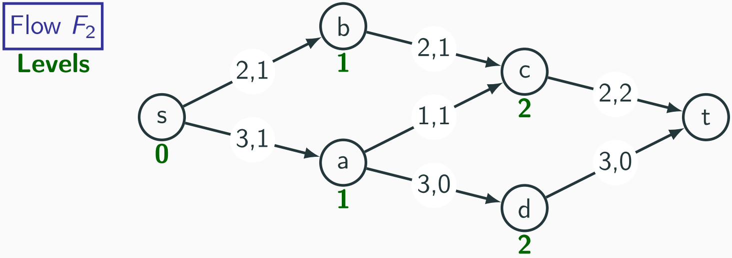
Vertex labelling algorithm, Example 3



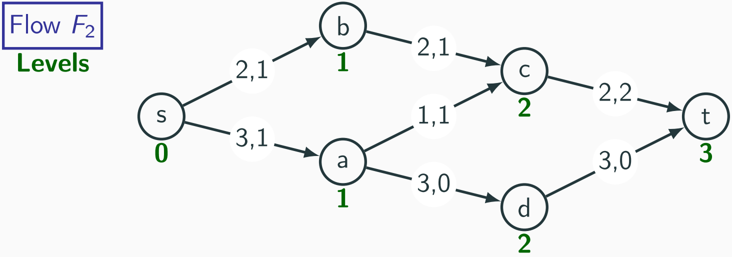
Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3

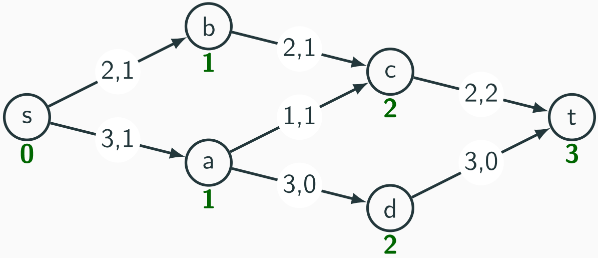


Vertex labelling algorithm, Example 3

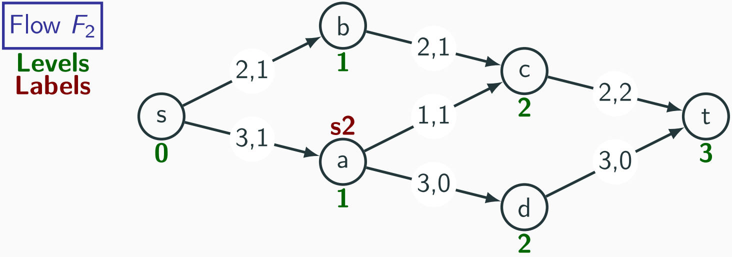
Flow F_2

Levels

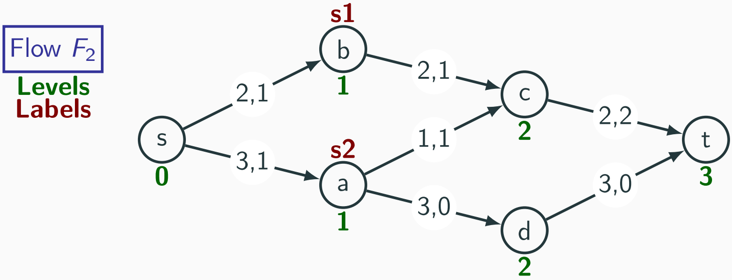
Labels



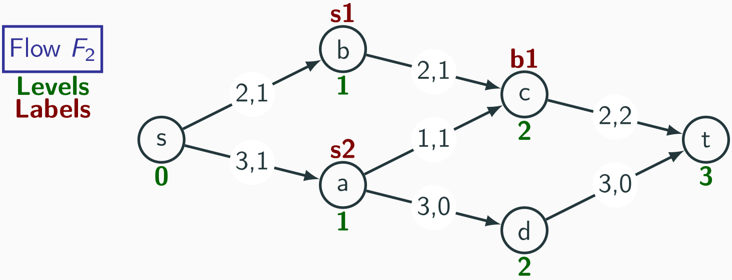
Vertex labelling algorithm, Example 3



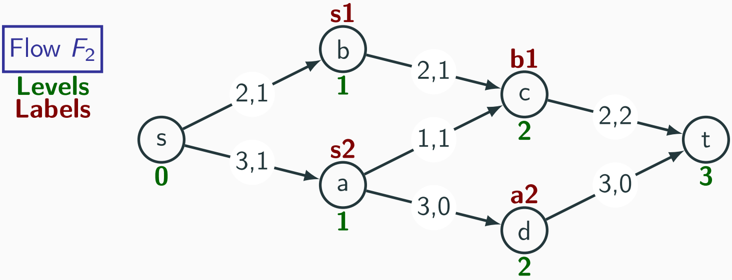
Vertex labelling algorithm, Example 3



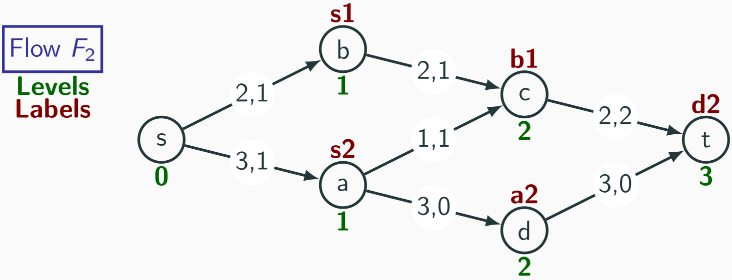
Vertex labelling algorithm, Example 3



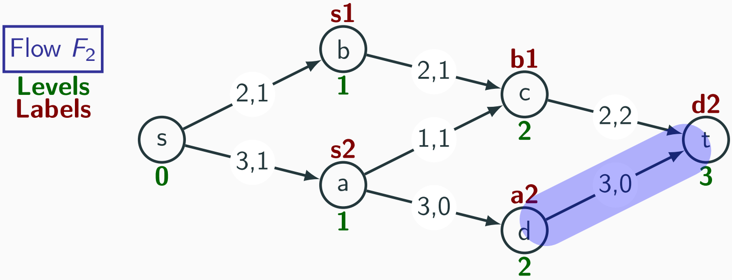
Vertex labelling algorithm, Example 3



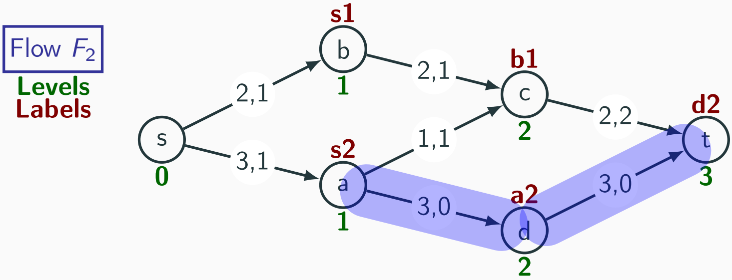
Vertex labelling algorithm, Example 3



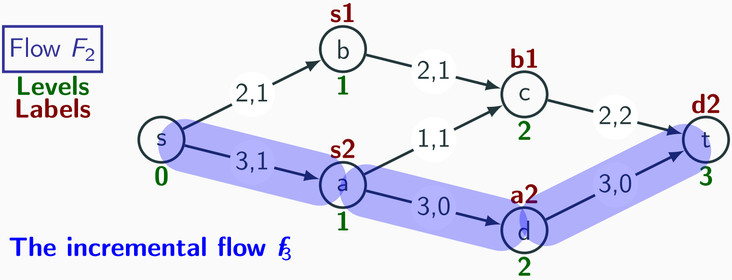
Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3

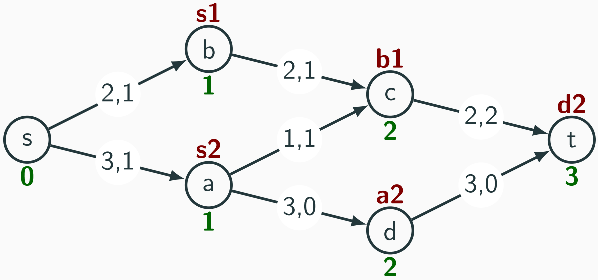


The incremental flow f_3

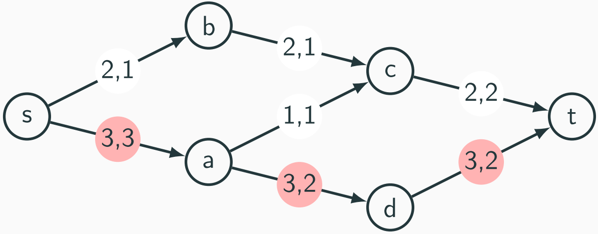
Vertex labelling algorithm, Example 3

Flow F_2

Levels
Labels

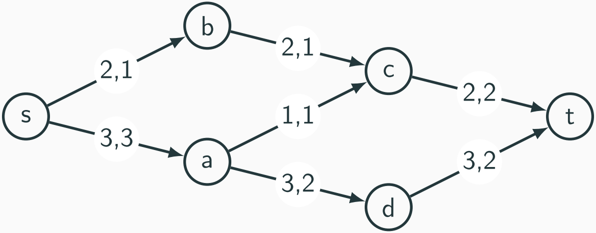


Flow F_3



Vertex labelling algorithm, Example 3

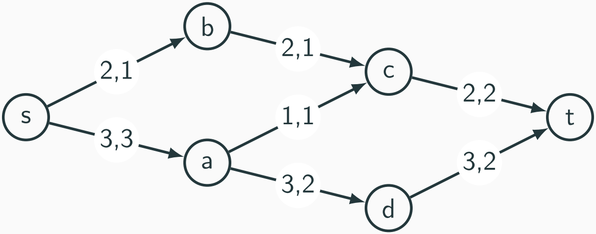
Flow F_3



Vertex labelling algorithm, Example 3

Flow F_3

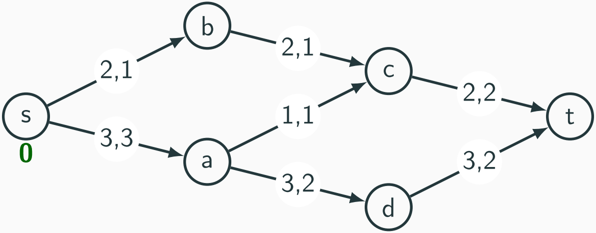
Levels



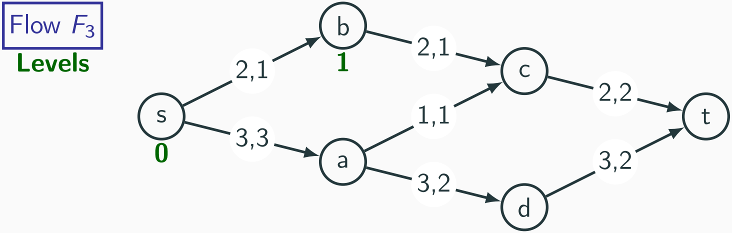
Vertex labelling algorithm, Example 3

Flow F_3

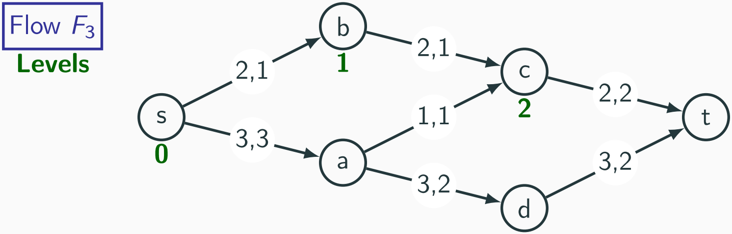
Levels



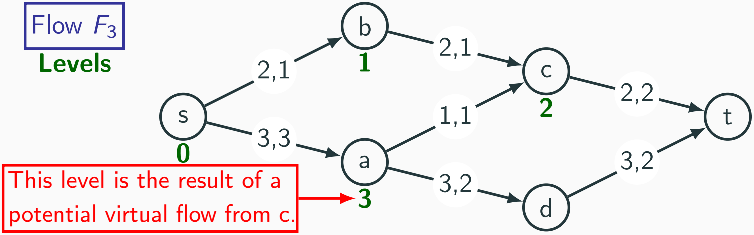
Vertex labelling algorithm, Example 3



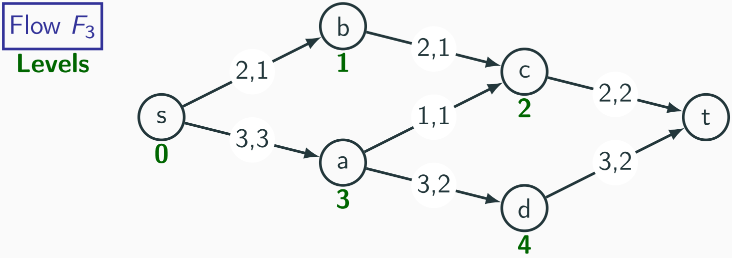
Vertex labelling algorithm, Example 3



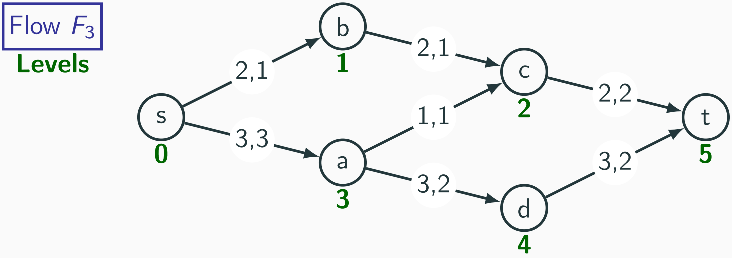
Vertex labelling algorithm, Example 3



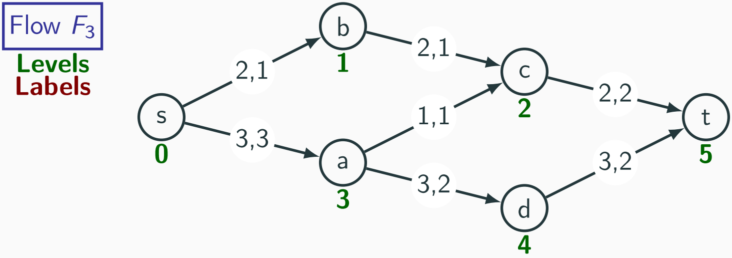
Vertex labelling algorithm, Example 3



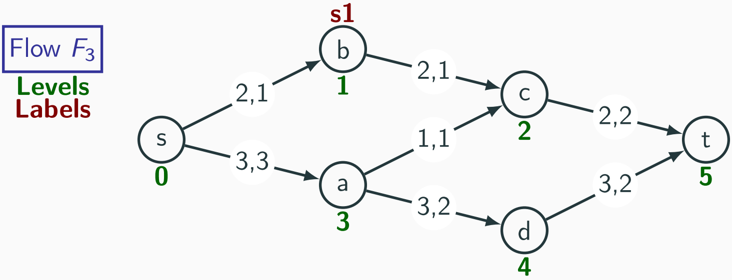
Vertex labelling algorithm, Example 3



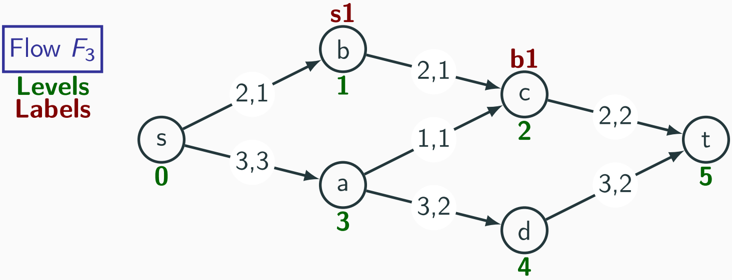
Vertex labelling algorithm, Example 3



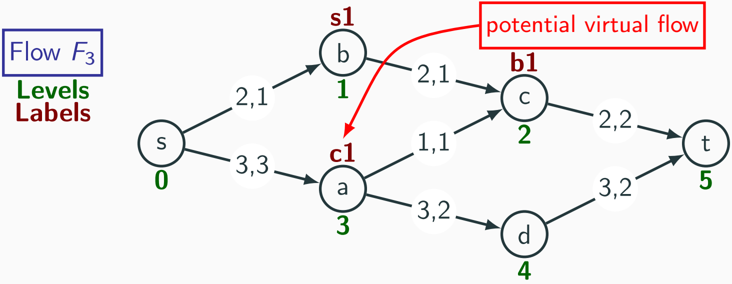
Vertex labelling algorithm, Example 3



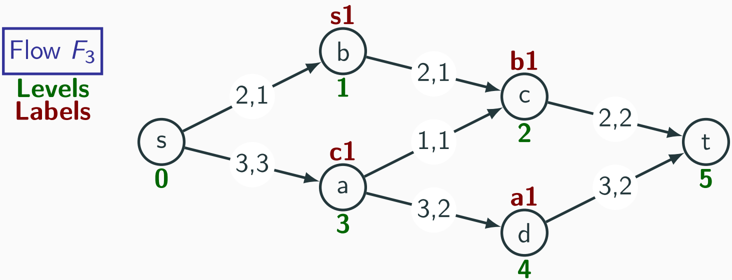
Vertex labelling algorithm, Example 3



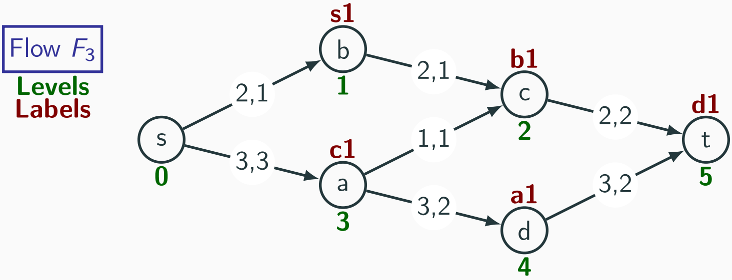
Vertex labelling algorithm, Example 3



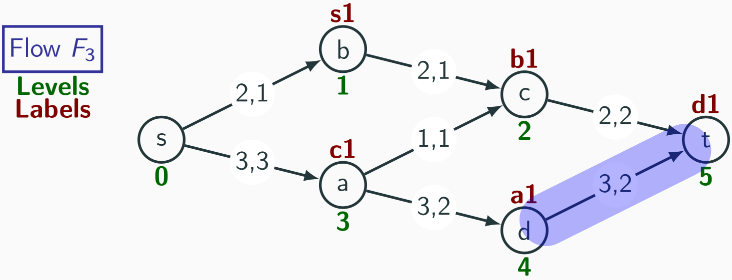
Vertex labelling algorithm, Example 3



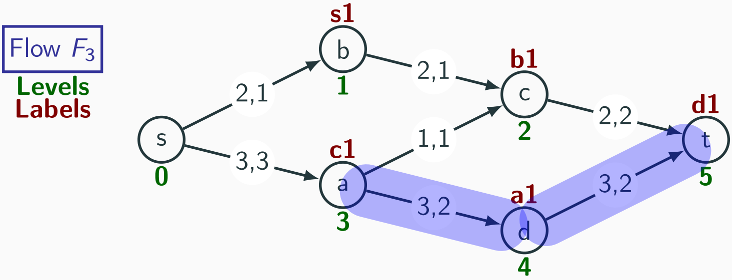
Vertex labelling algorithm, Example 3



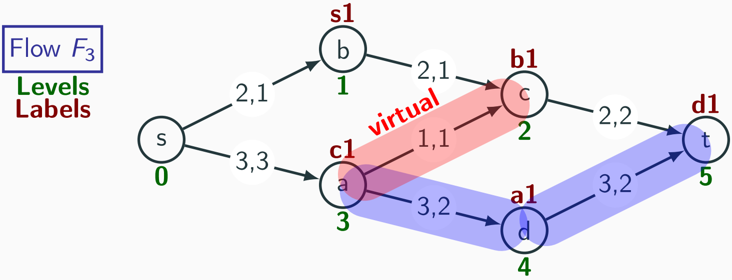
Vertex labelling algorithm, Example 3



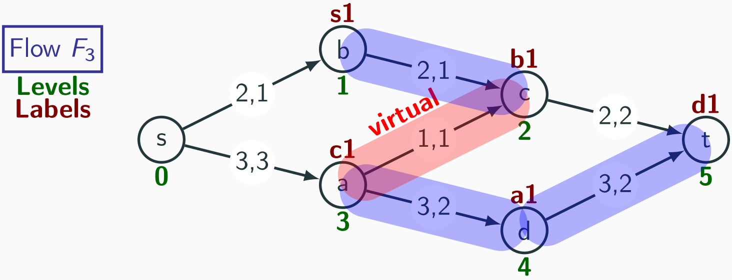
Vertex labelling algorithm, Example 3



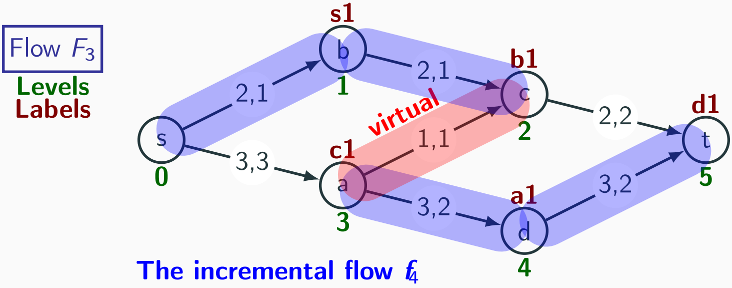
Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3



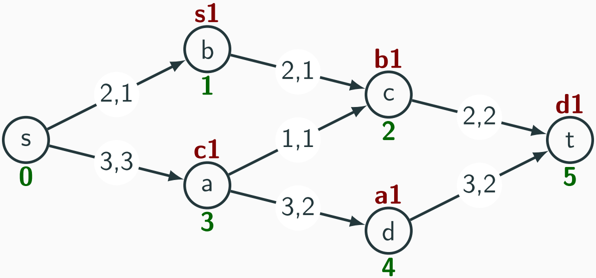
Vertex labelling algorithm, Example 3



Vertex labelling algorithm, Example 3

Flow F_3

Levels
Labels



Flow $F_4 = F_{\max}$