15.082J and 6.855J and ESD.78J

November 4, 2010

The Network Simplex Algorithm

Overview first 1/3 of the lecture

Quick Review of Linear Programming:

- some geometry
- extreme points (corner points)
- an overview of the simplex algorithm

Networks

- extreme points
- basic feasible solutions

The Network Simplex Algorithm

A Two Variable Linear Program

objective $z = 3x + 5y$

$2x + 3y \leq 10$ (1)

$x + 2y \leq 6$ (2)

$x \leq 4$ (3)

$y \leq 3$ (4)

$x, y \geq 0$ (5)

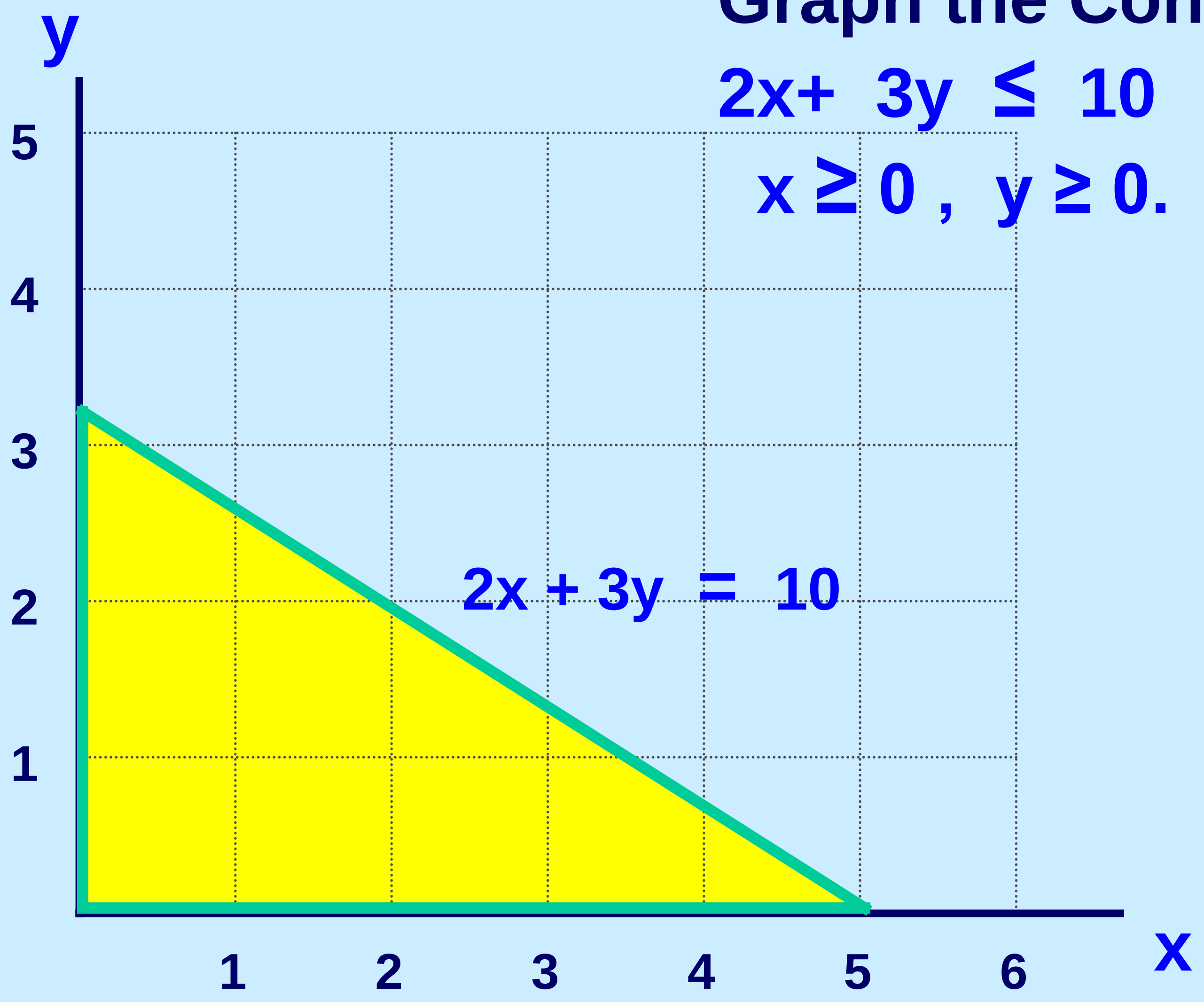
Constraints

Graphing the Feasible Region

Graph the Constraints:

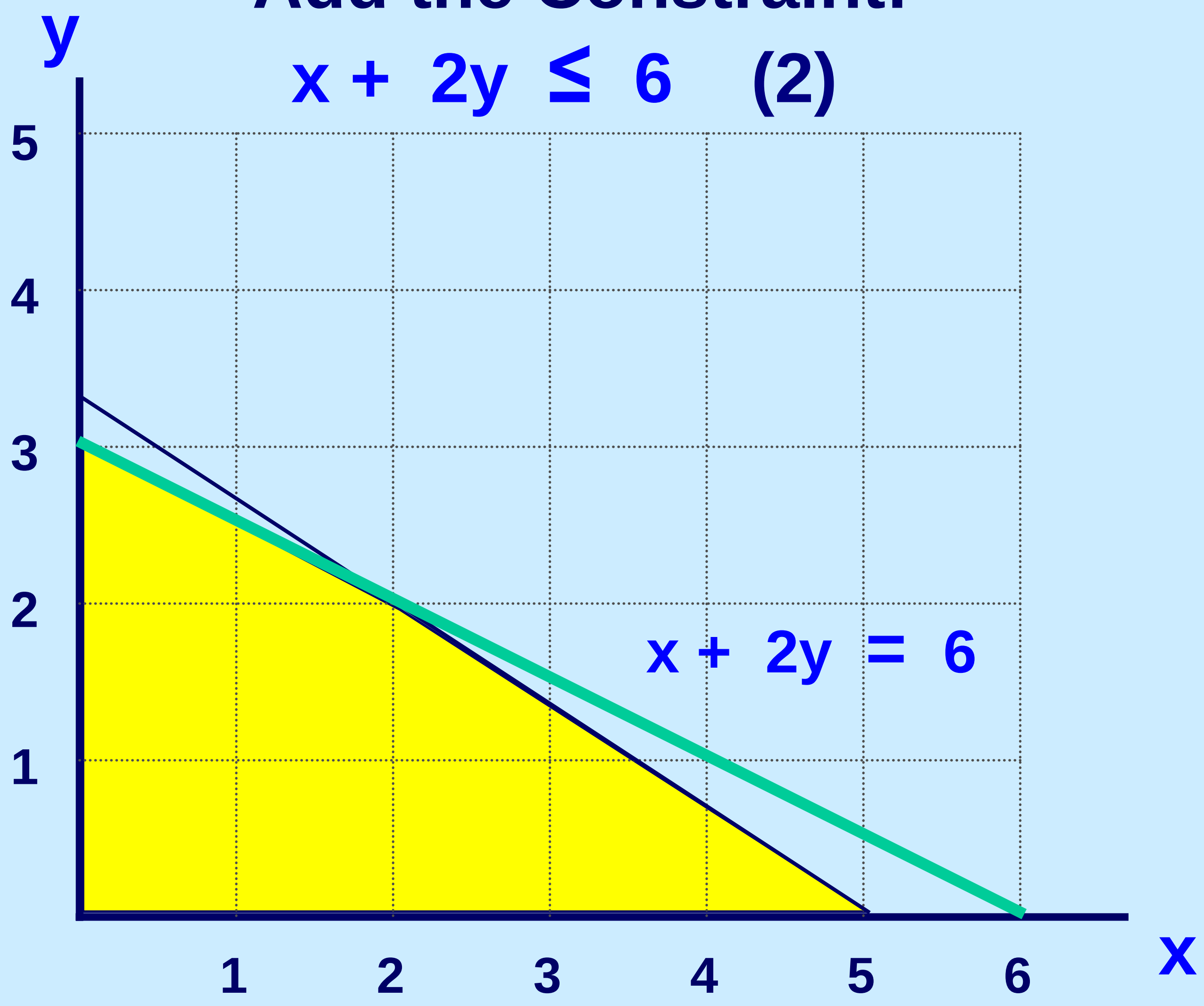
$2x + 3y \leq 10$ (1)

$x \geq 0, \quad y \geq 0.$ (5)



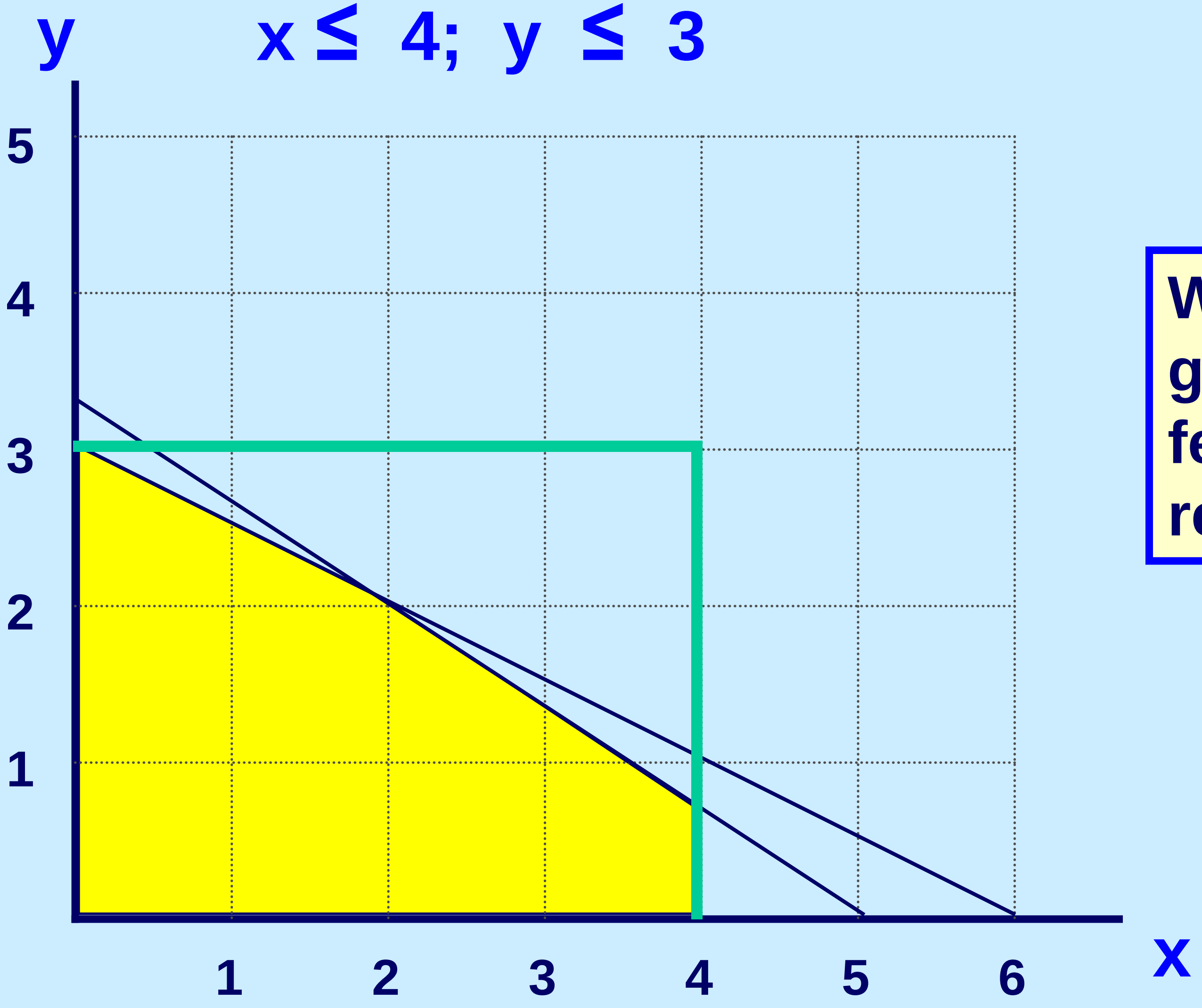
Add the Constraint:

$x + 2y \leq 6 \quad (2)$



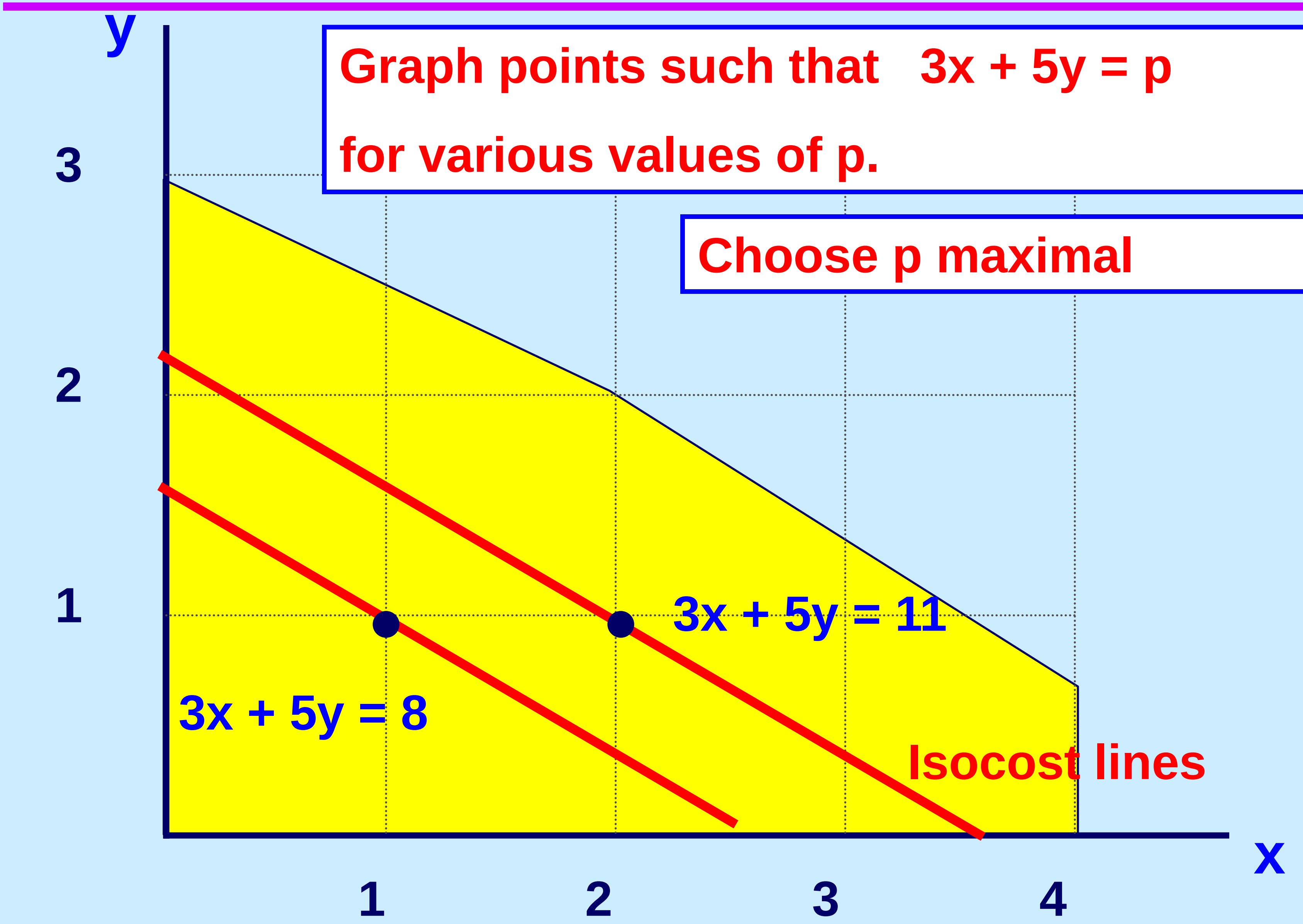
Add the Constraints:

$x \leq 4; \;\; y \leq 3$



We have now graphed the feasible region.

The geometrical method for optimizing 3x + 5y



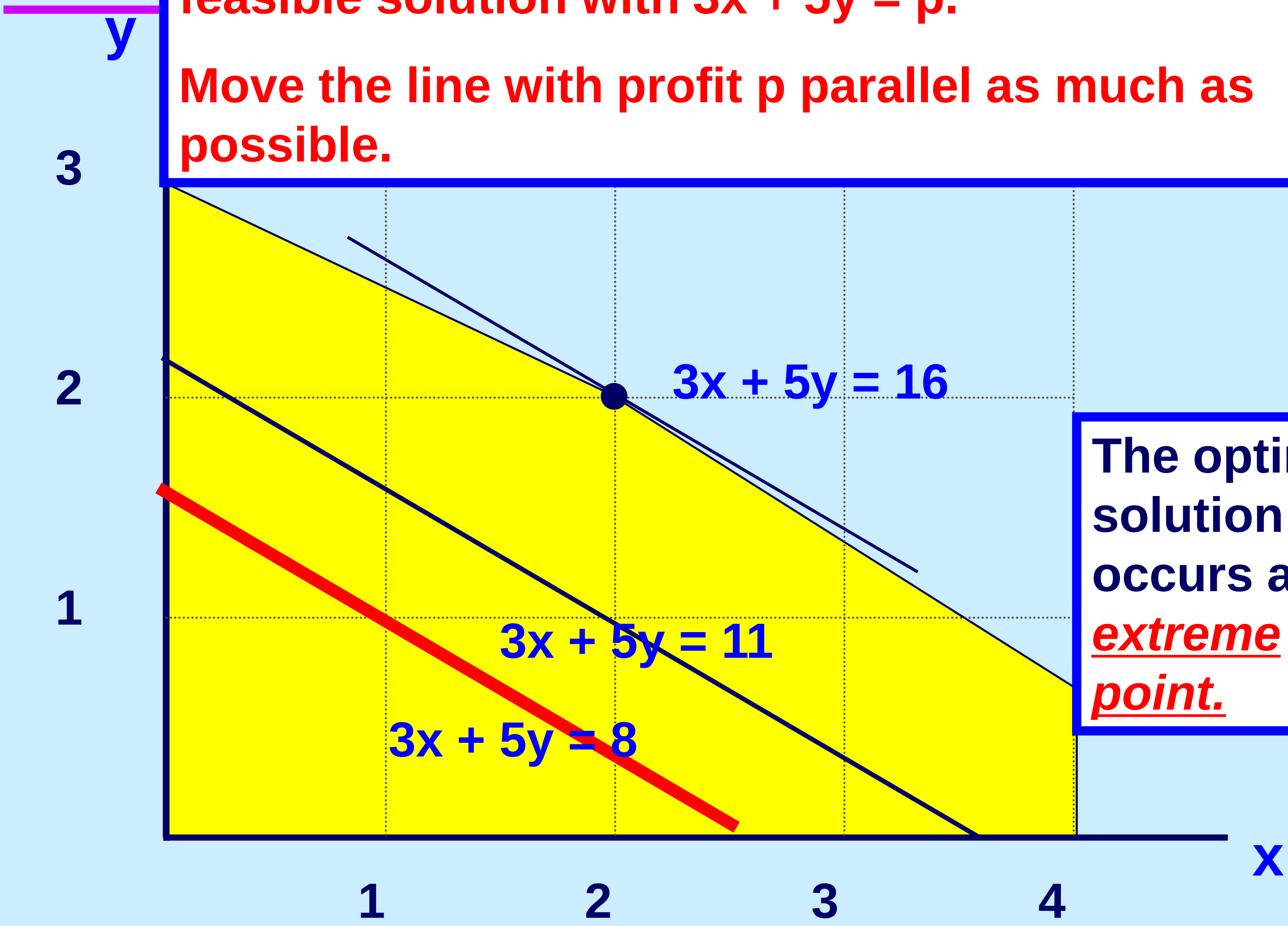
Graph points such that 3x + 5y = p

for various values of p.

Choose p maximal

Find the maximum value p such that there is a feasible solution with 3x + 5y = p.

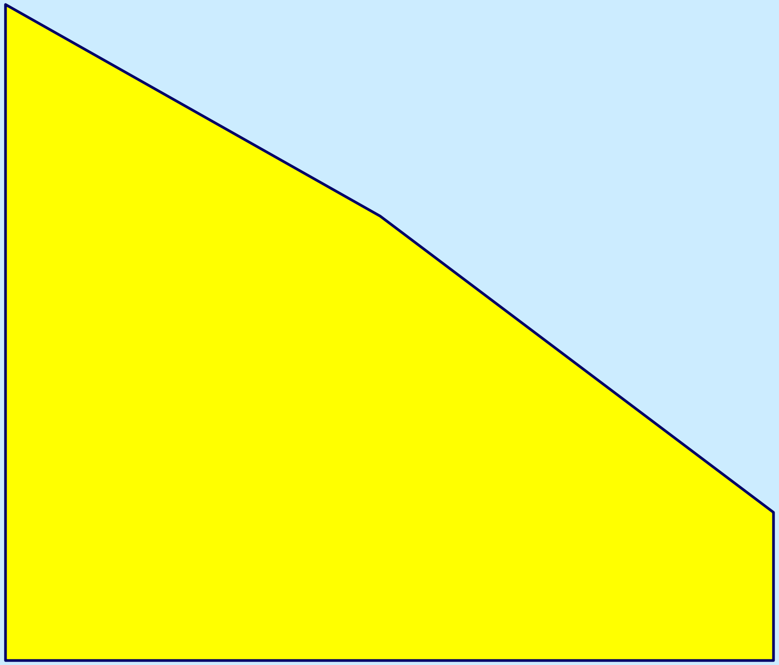
Move the line with profit p parallel as much as possible.



The optimal solution occurs at an ***extreme point.***

Extreme Points (Corner Points)

An ***extreme point*** (also called n ***corner point***) of the feasible region is a point that is not the midpoint of two other points of the feasible region.



Where are the extreme points of this feasible region?

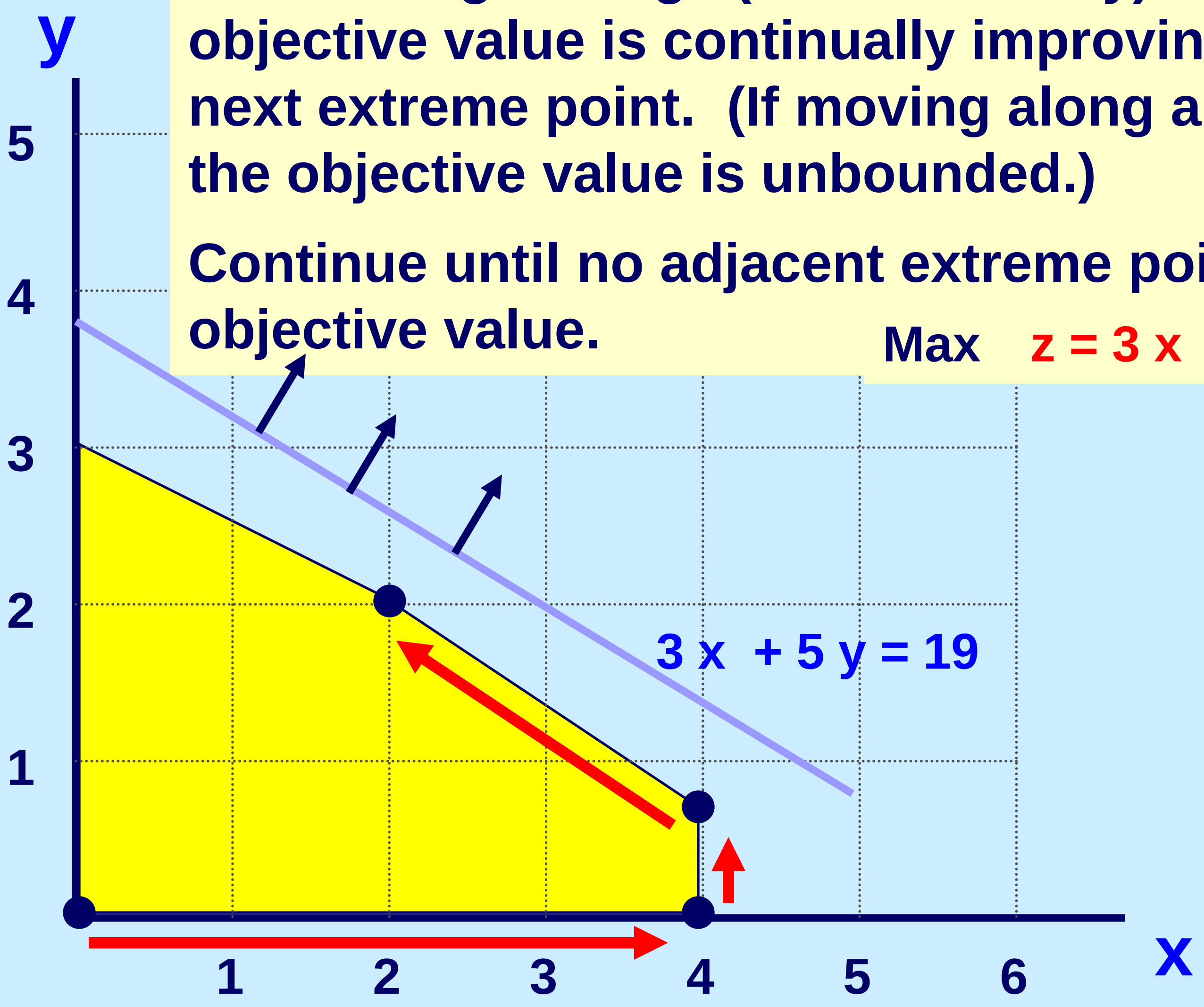
The Simplex Method

Start at any feasible extreme point.

Move along an edge (or extreme ray) in which the objective value is continually improving. Stop at the next extreme point. (If moving along an extreme ray, the objective value is unbounded.)

Continue until no adjacent extreme point has a better objective value.

Max $z = 3x + 5y$



Comments about the simplex algorithm

Each step is called a pivot.

- Pivots are carried out using linear algebra
- Pivots for network flow problems can be carried out directly by changing flows in arcs.

Typically, the simplex method finds the optimal solution after a “small” number of pivots (but can be exponential in the worst case).

The simplex algorithm is VERY efficient in practice.

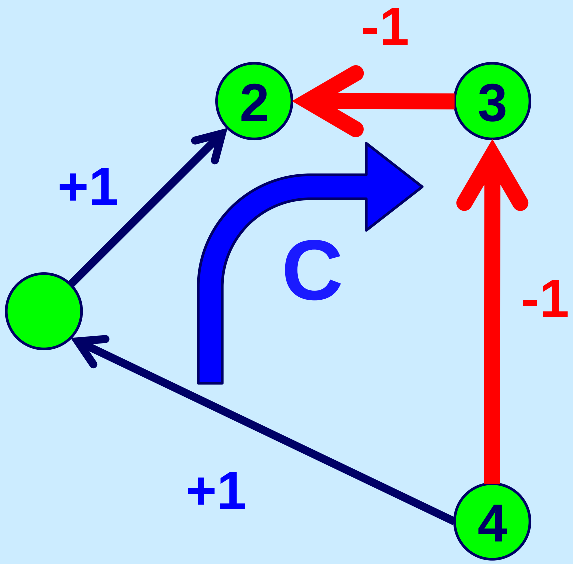
Back to networks

We will work with the arcs of the original network (not the residual network)

We will soon describe extreme flows.

Connection between spanning tree flows and extreme flows.

Networks: sending flow around a cycle



Assume that cycles are oriented in a direction. The **forward arcs** of the cycle are the arcs in the same orientation. The **backward arcs** are in the opposite direction.

Arcs (1,2) and (4, 1) are forward arcs.
Arcs (3, 2) and (4, 3) are backward arcs.

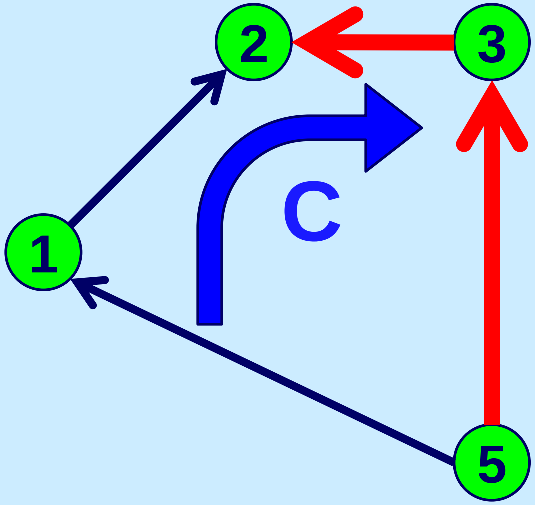
A flow of 1 unit around C refers to a flow of 1 unit in the forward arcs and a flow of -1 units in the arcs in the backward arcs.

What is an extreme point solution of a network flow problem?

Let x be a feasible flow for a minimum cost flow problem. An arc (i, j) is called free if $0 < x_{ij} < u_{ij}$. One can increase or decrease the flow in a free arc by a small amount and still satisfy bound constraints.

Theorem. A feasible flow x is an extreme point solution if and only if there is no (undirected) cycle of free arcs.

Proof. Suppose that C is a free cycle. We will show that x is not extreme. Let y_C be a flow of 1 unit around C.



Since all arcs of C are free wrt x, there is some $\varepsilon > 0$ so that $x' = x + \varepsilon\, y_C$ is a feasible flow and so that

$x'' = x - \varepsilon\, y_C$ is feasible.

x is the midpoint of x' and x".

Proof continued

Suppose that x is not an extreme point. Suppose that
x = (x’ + x”)/2 and for feasible flows x’ and x”.

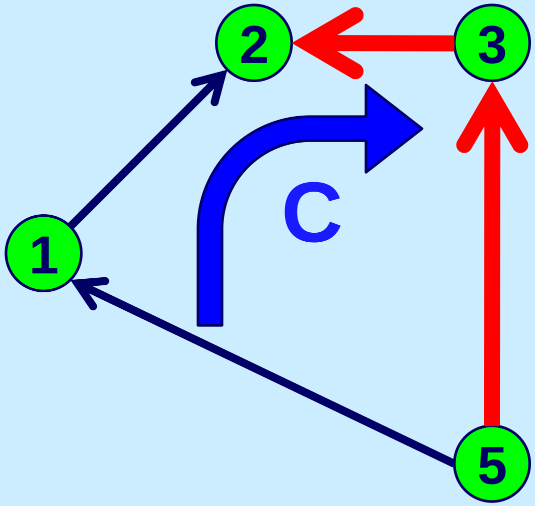
We will show that there is a free cycle.

Consider arc (i, j). $x_{ij} = (x'_{ij} + x''_{ij})/2$.

if $x_{ij} = 0$ then $x'_{ij} = x''_{ij} = 0$ (Otherwise $x'_{ij} < 0$ or $x''_{ij} < 0$).

If $x_{ij} = u_{ij}$ then $x'_{ij} = x''_{ij} = u_{ij}$ (Otherwise $x'_{ij} > u_{ij}$ or $x''_{ij} > u_{ij}$).

So, x differs from x’ and x” on free arcs.



x’ – x is a circulation, and so it is expressible as the sum of flows around cycles. It follows that there is a cycle of free arcs.

Basic feasible solutions

A **basis structure** consists of a spanning tree T, a set L of arcs, and a set U of arcs, such that T∪ L ∪ U = A.

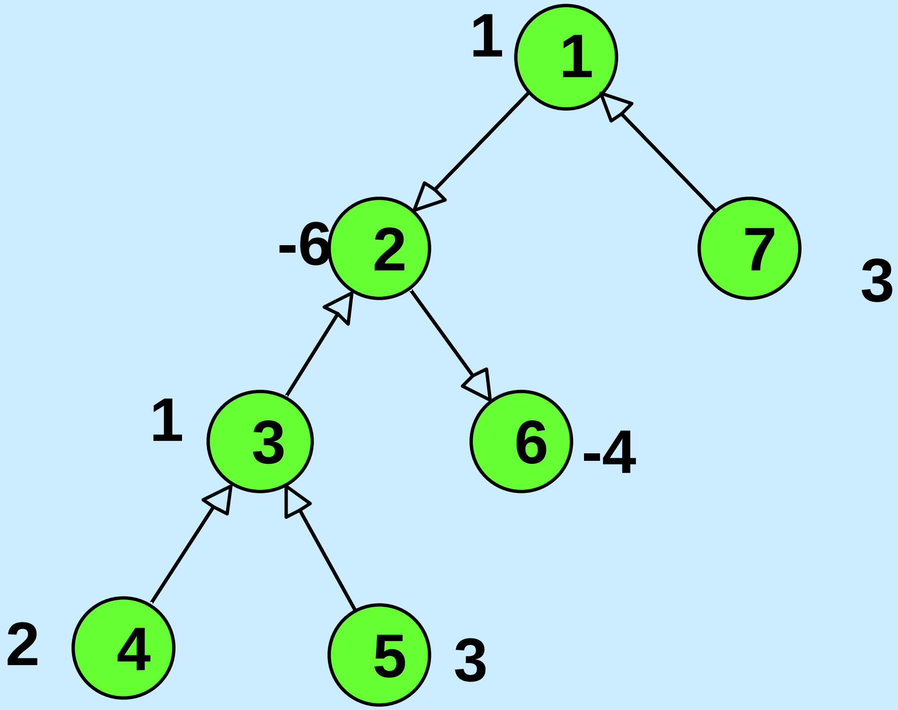
For each (i,j) ∈ L, $x_{ij} = 0$.

For each (i,j)∈ U, $x_{ij} = u_{ij}$.

The arc flows in T are selected so that each node satisfies its supply/demand constraint.

The basis structure is **feasible** if the arc flows also satisfy the upper and lower bounds. (Not all basis structures are feasible.) The feasible flow is called **basic**.

Calculating A Spanning Tree Flow

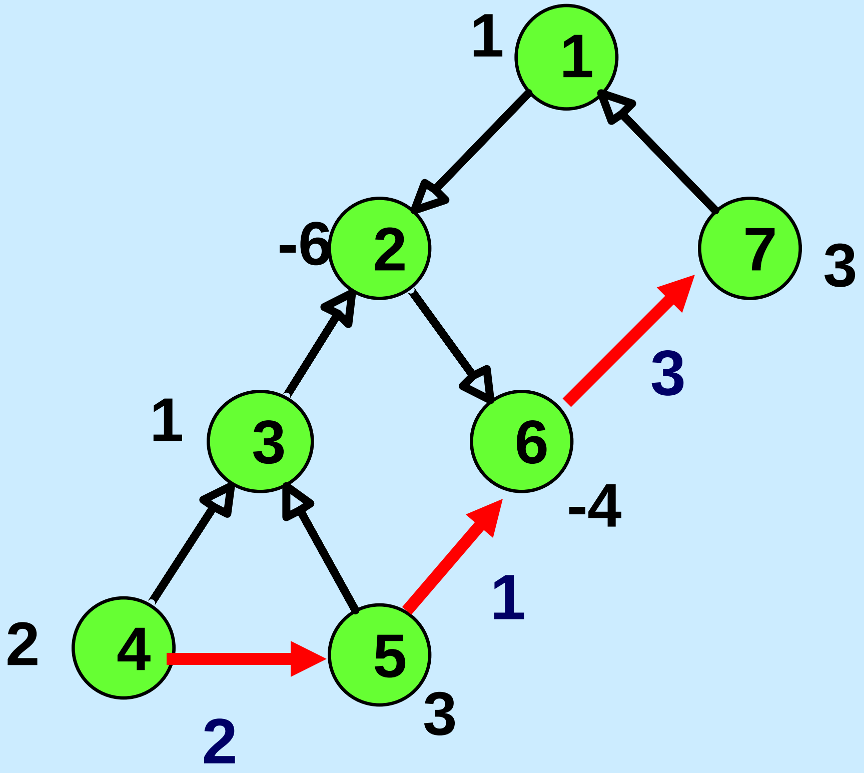


A tree with supplies and demands. (Assume that all other arcs have a flow of 0)

What is the flow in arc (4,3)?

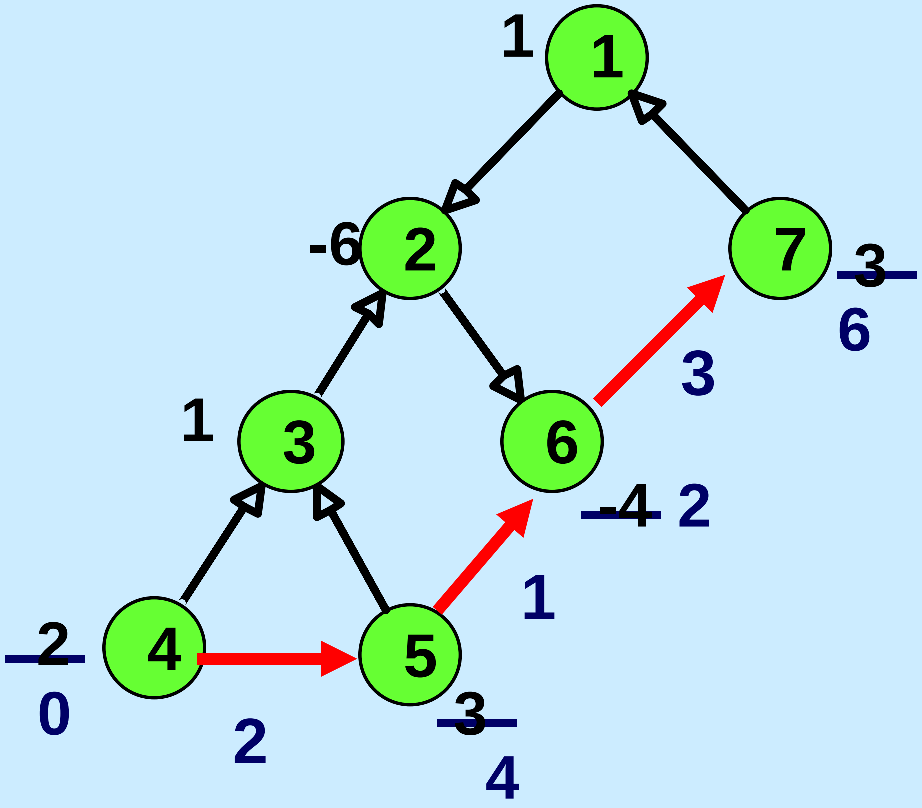
See the animation: slide 2

What would happen if the flows in non-tree arcs were not 0?



Suppose that non-tree arcs had a non-zero flow. How would this change the computations?

What would happen if the flows in non-tree arcs were not 0?



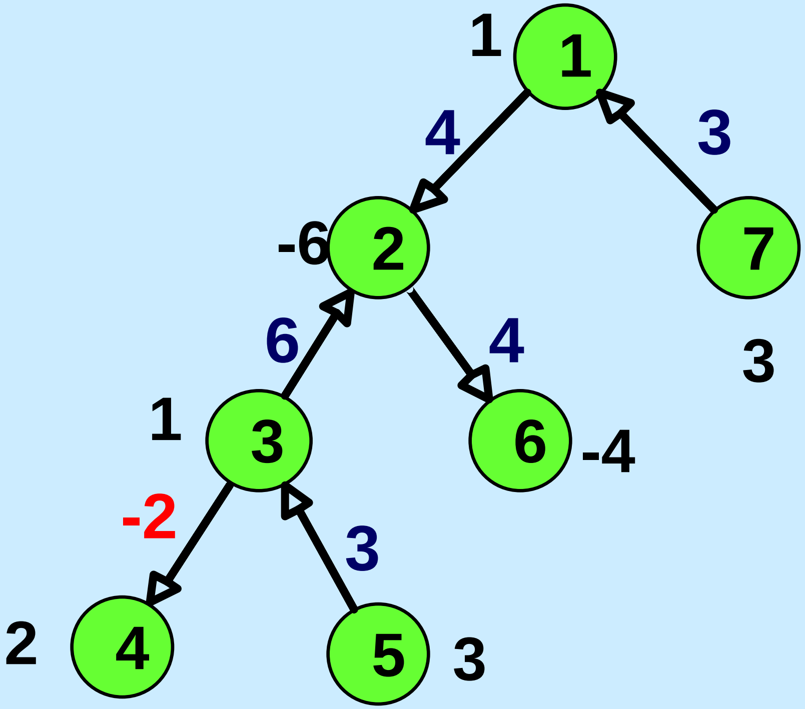
Adjust the supplies/demands.

They will be interpreted as excesses and deficits.

The compute flows as in the previous method; e.g., what is the flow in (4,3)?

What would happen if the flow were negative?

If the direction of (4,3) were reversed, the flow in (3,4) would be negative.

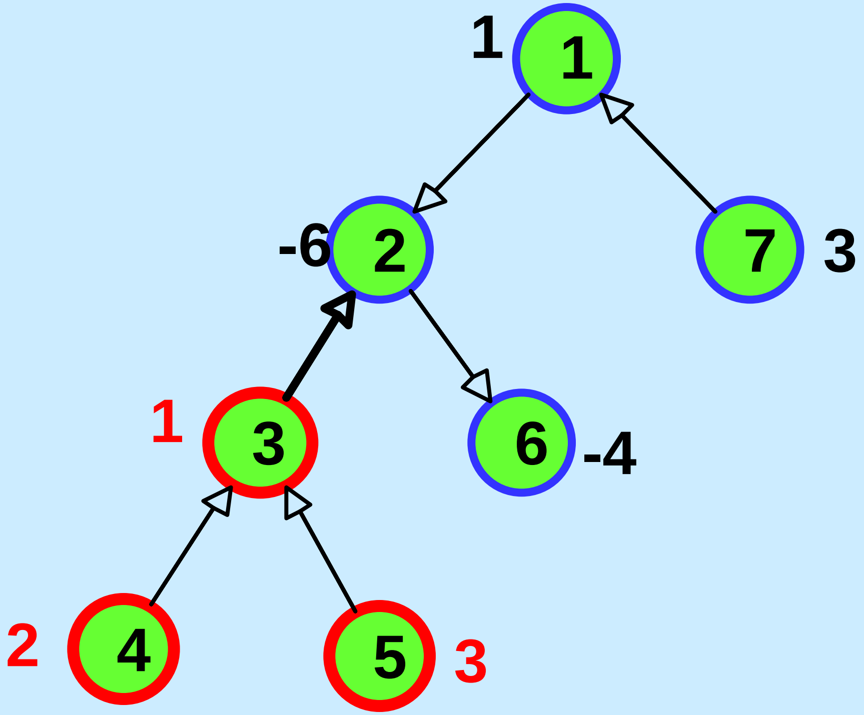


A spanning tree flow **is guaranteed** to satisfy the supply/demand constraints. It may violate an upper or lower bound.

A spanning tree flow is called **feasible** if it satisfies its upper and lower bound. Otherwise, it is **infeasible**.

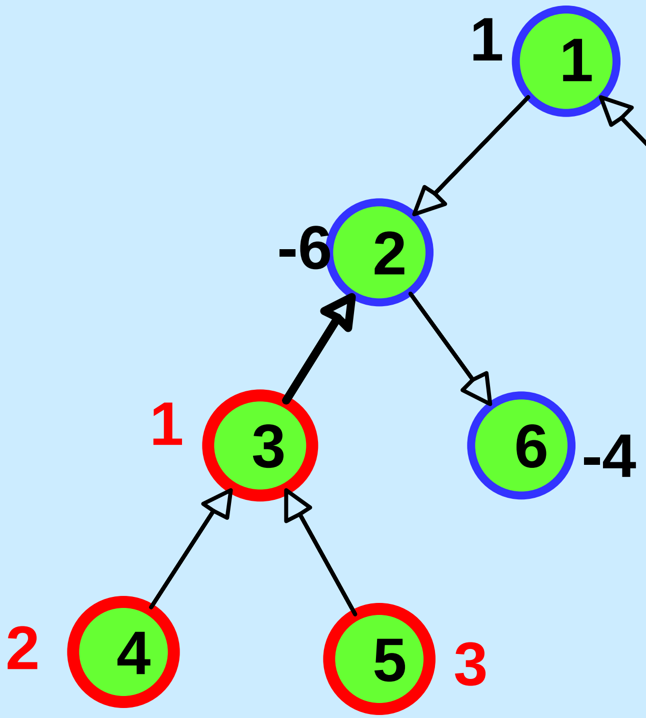
Another way of calculating flows in arcs

Case 1. If (i, j) is not in the tree, then $x_{ij} = 0$.



The total supply in subset S = (3,4,5) of nodes is 6. How can one satisfy the supply/demand constraints for S?

Another way of calculating flows in arcs

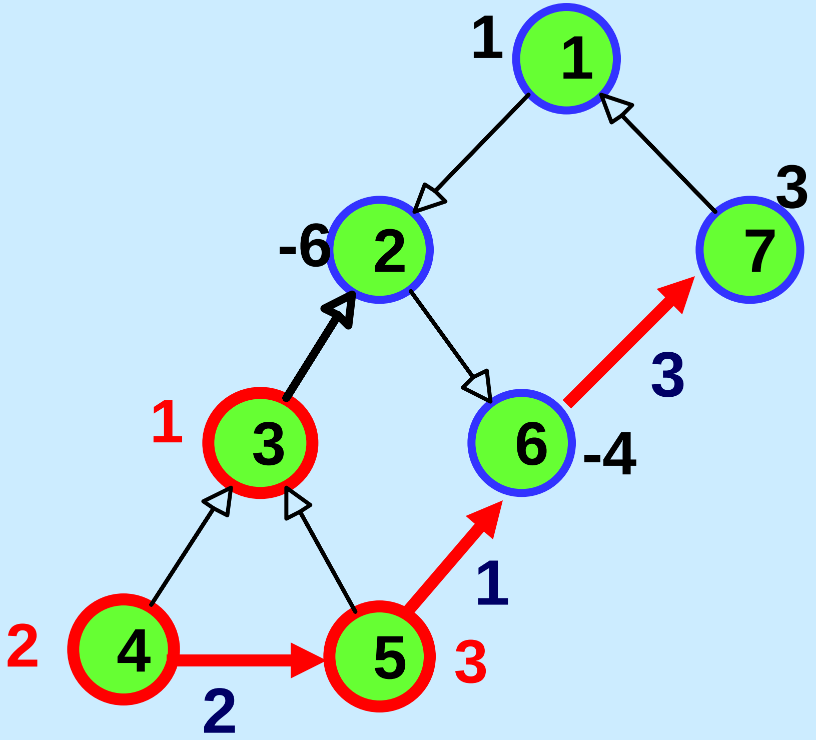


Deleting an arc (i,j) of T splits the nodes into two subsets, S and N-S.

To compute the flow in (i,j), compute $\sum_{j \in S} b(j)$.

Another way of calculating flows in arcs, general case

Case 2. If (i,j) is not in the tree, then x_{ij} = 0 or u_{ij}



Deleting an arc (i,j) of T splits the nodes into two subsets, S and N-S.

Let f(S, N-S) denote the flow across the cutset (S, N-S) from the non-tree arcs.

To compute the flow in (i, j), compute

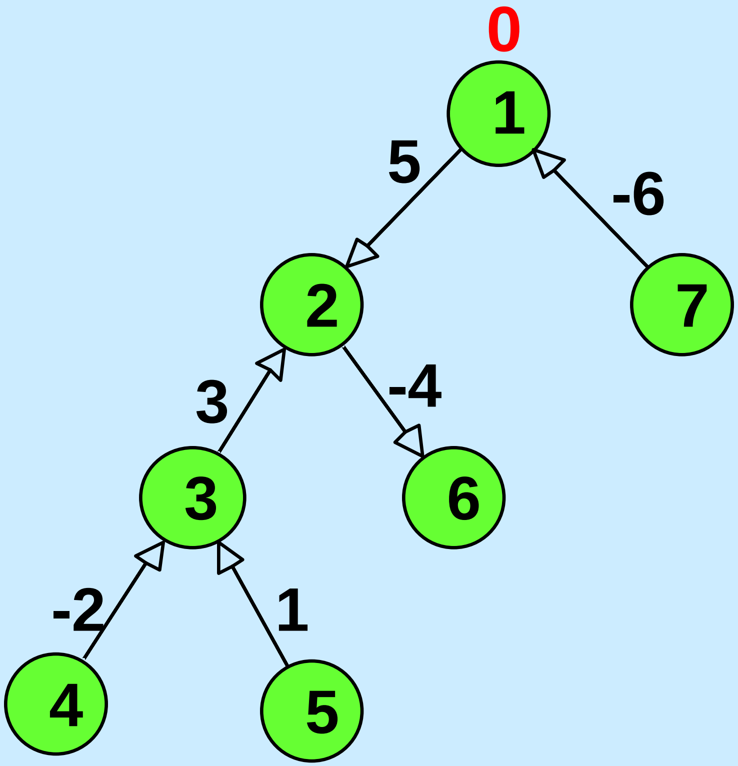
$\sum_{j \in S} b(j) \quad - \quad f(S, N\text{-}S)$.

Simplex Multipliers

Simplex multipliers for the network simplex algorithm are a special case of node potentials.

- They are selected so that the reduced costs of every tree arc is 0.
- In the simplex algorithm for linear programs, basic variables have a reduced cost of 0.

Calculating Simplex Multipliers for a Spanning Tree



To calculate node potentials,

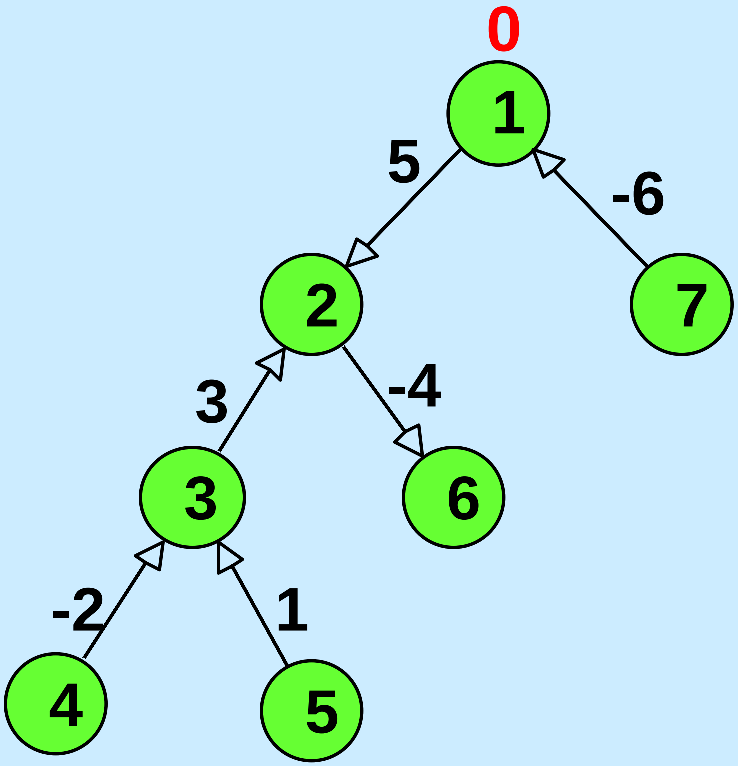
1. Let $\pi_1 = 0$;

2. Choose other multipliers so that for each arc (i,j) in the tree $c_{ij} - \pi_i + \pi_j = 0$.

What is the node potential for node 2?

See the animation page 9.

An alternative approach for calculating simplex multipliers



Let π_i be the cost of the path from node i to node 1 (the root node) in T. If (j,k) is backward, then use cost $-c_{jk}$.

What is the simplex multiplier for node 4?

What is the simplex multiplier for node 6?

Mental Break

How tall is a baby giraffe at birth?

Around 6 feet

In 1859, 24 rabbits were released in Australia. How many were there 6 years later?

More than 2 million.

The female American oyster lays lots of eggs per year even though only one of the bunch reaches maturity. Approximately how many eggs does the oyster lay?

About 500 million

Mental Break

Human beings typically mate face-to-face. How many other kinds of land animals mate face-to-face.

One other: the 2-toed sloth.

Humans have around 600 muscles. How many muscles do caterpillars have?

Around 6000

What do bats do upon exiting a cave?

They turn left.

Optimality Conditions (again)

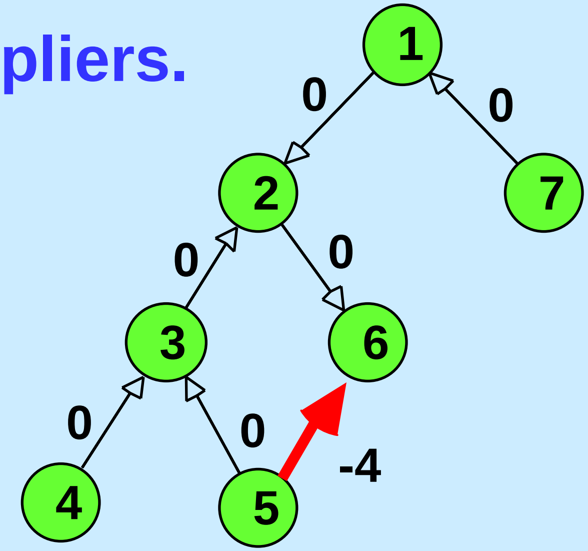
Optimality Conditions for Spanning Tree Solutions:

The following are conditions under which x is an optimal solution for the minimum cost flow problem and π is optimal for the dual problem:

1. The basic flow x is feasible
2. π is the vector of simplex multipliers.
3. For each non-tree arc (i, j)

   a. if $c^{\pi}_{ij} > 0$, then $x_{ij} = 0$

   b. if $c^{\pi}_{ij} < 0$, then $x_{ij} = u_{ij}$



What is the flow on arc (5,6) if arc (5,6) satisfies the optimality conditions?

Violating Arcs

(T, L, U) : a spanning tree structure.

x: basic feasible flow

π : simplex multipliers.

c^{π}_{ij} reduced costs

A non-tree arc is a ***violating arc*** and eligible for entering the basis if

i. $c^{\pi}_{ij} < 0$ and $x_{ij} = 0$ or

ii. $c^{\pi}_{ij} > 0$ and $x_{ij} = u_{ij}$.

The Network Simplex Pivot

1. Choose a violating non-tree arc. If no such arc exists, then the solution x is optimal.

2. Add (i,j) to T creating a unique cycle C. Send a maximum flow around C while maintaining feasibility. Suppose the exiting arc is (p,q).

3. Update the multipliers so that the reduced costs of all tree arcs are 0 after the pivot.

T-(p,q) partitions into two subtrees, T_1 and T_2 with the root node in T_1.

Let d = $|c^{\pi}_{ij}|$.

If i $\in T_1$, then add $|c^{\pi}_{ij}|$ to each node v$\in T_2$.

If i $\in T_2$, then subtract d from each node in T_2.

The Steps for the network simplex

1. Select the entering arc.

   Key data structure: maintain the simplex multipliers.

   O(1) step to determine if (i,j) is violating.

2. Determine the basic cycle C. Determine the flow around the basic cycle. Send the flow.

3. Determine the subtree T_2 obtained upon deleting the exiting arc from the current spanning tree. Update all multipliers in T_2.

See page 17 of animation

On implementations

The time it takes is sensitive to the data structures used to implement the algorithm.

All reasonable implementations take O(n) time per pivot.   Some work slightly better in practice (as described in the text.)

Some Remaining Issues

How can we avoid cycling in the simplex method? (Or what do we do if the amount of flow sent around a cycle is 0).

What is the worst case performance of the simplex method?

What are some good heuristics to speed up performance in practice?

Summary

1. **Network simplex is extremely fast in practice.**

2. **Relying on network data structures, rather than matrix algebra, causes the speedups. It leads to simple rules for selecting the entering and exiting variables.**

3. **A good pivot rule can dramatically reduce running time in practice.**

Optional Material: Data Structures for the Network Simplex Algorithm

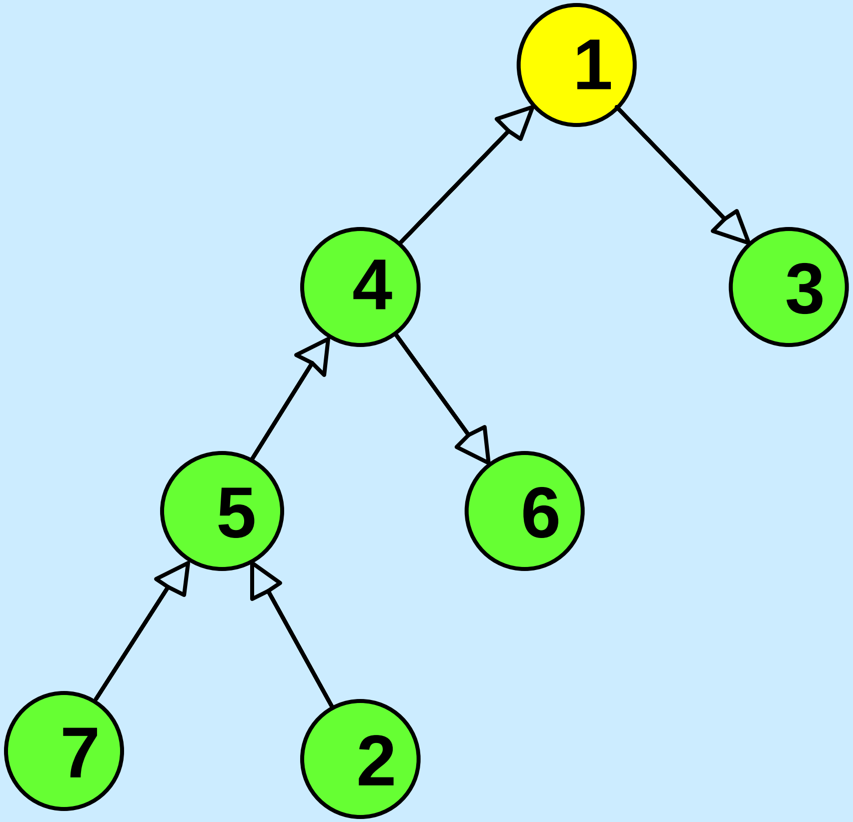
1. Hang the tree from a root node.

2. For each node, store pred(i), the parent of i in the tree.

3. For each node i store the depth(i), the number of arcs on the path from i to the root.

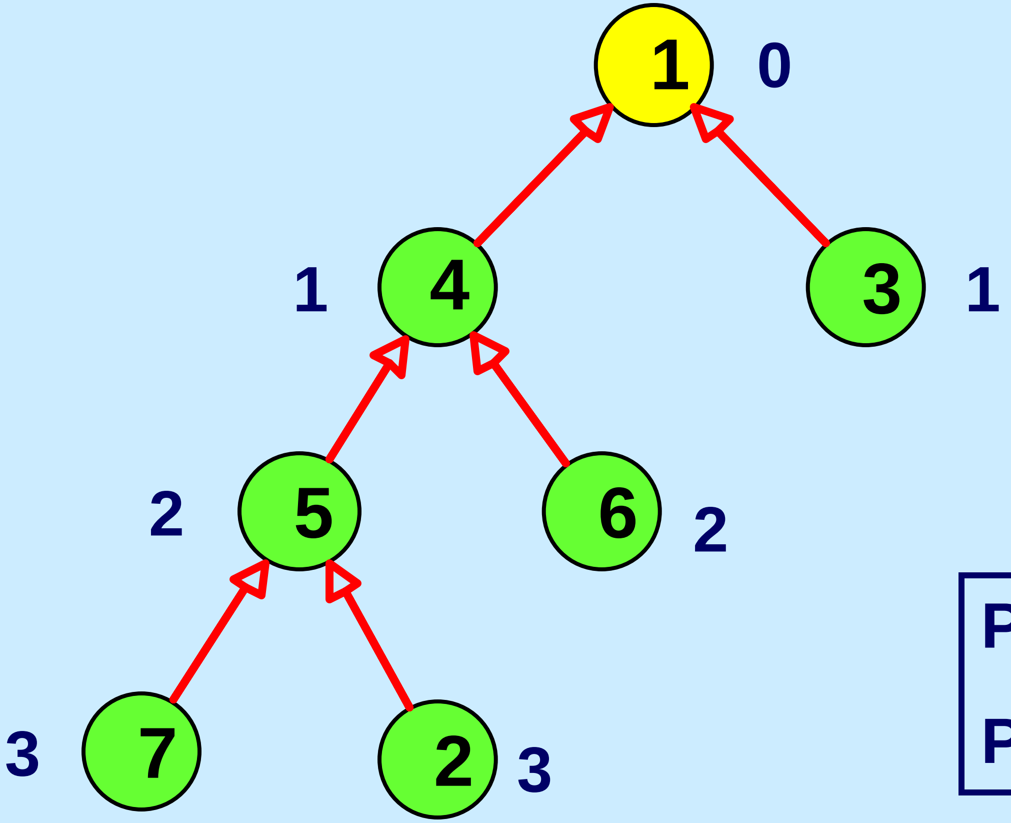
4. For each node i, store a pointer to the next node on the depth first search ordering of T. This data structure is called the *thread*.

A spanning tree, as part of a spanning tree solution



Node 1 is the root node. The tree “hangs” from the root node.

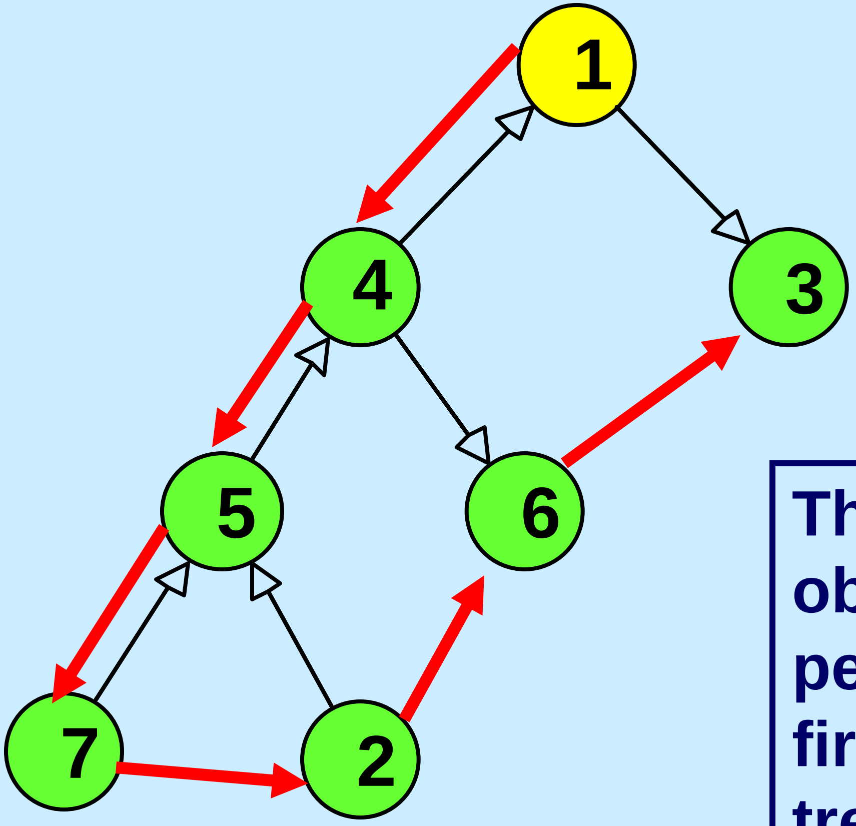
The Depth of Nodes, and Predecessors



Pred(2) = 5,

Pred(3) = 1, etc.

The Thread



The **thread** is obtained by performing a depth first search of the tree. Each node points to the next one on the dfs.

Finding the Cycle

```
procedure  IDENTIFY CYCLE;
begin
    set i : = k and j : = l;
    while  i ≠ j do
    begin
        if depth(i) > depth(j) then i : = pred(i)
        else if depth(j) > depth(i) then j : = pred(j)
            else i : = pred(i) and j : = pred(j);
    end;
    set w : = i;
end;
```

Running time is O(|C|)

Updating the Multipliers

Suppose: (i,j) is pivoted in and that (p,q) is pivoted out.

Suppose that T_2 consists of node p plus the descendents of p in the tree.

Use the thread to trace out T_2.

Stop when depth(CurrentNode) ≤ depth(p).

Running time = O(| T_2|).

Updating the data structures. O(|C| + | T_2|).