proof Lemma 1.56:

(a) "⇒": If there is a set $X \subset V(G)$ with $r \in X$, $v \in V(G) \setminus X$ and $\delta(X) = \emptyset$, there can be no r-v-path, so G is not connected.

"⇐": If G is not connected, there is no r-v-path for some r and v.
Let R be the set of vertices reachable from r.
We have $r \in R$, $v \notin R$ and $\delta(R) = \emptyset$.

(b) analogously

□

proof Lemma 1.60

Let $e = (x, y)$. We label the vertices of G by the following procedure:

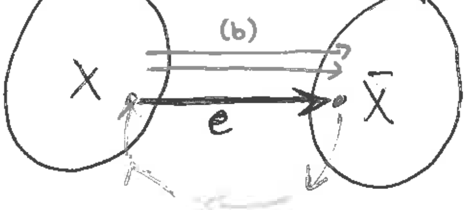
First label y.
In case v is already labelled and w is not, we label w if there is a **black** edge (v,w), a red edge (v,w) or a red edge (w,v).
In this case, we write $pred(w) := v$.

When the labelling procedure stops, there are two possibilities

(1): <u>x has been labelled</u>. Then the vertices $x, pred(x), pred(pred(x)), \ldots, y$ form an undirected circuit with the properties (a).

(2): <u>x has not been labelled</u>. Then let R consists of all labelled vertices. Obviously, the undirected cut $\delta^+(R) \cup \delta^-(R)$ has the properties (b).

Suppose that an undirected circuit C as in (a) and an undirected cut $\delta^+(X) \cup \delta^-(X)$ as in (b) both exist. All edges in their (nonempty) intersection are black, they all have the same orientation w.r.t. to C, and they all leave X or all enter X ↯ contradiction



Examples: Θ, O, Ω

Bsp.: • $2n^2 - 1 \in \Theta(n^2)$

$c_1 = 1$ $\quad 0 \leq c_1 n^2 \leq 2n^2 - 1 \leq 2n^2 \quad \forall n \geq n_0 = 1$

$c_2 = 2$ $\quad 0 \leq 1 n^2 \leq \cancel{2}n^2 + 1 - 1 \leq 2n^2 - 1 \leq 2n^2$

$n_0 = 1$

• $2n^2 - 1 \in O(n^2)$

$\in O(n^3)$ $\quad c_2 = 2, n_0 = 1$

$0 \leq 2n^2 - 1 \leq 2n^3 \quad \forall n \geq n_0 = 1$

• $n \log n \in O(n^2)$

$c_2 = 1$ $\quad 0 \leq n \cdot \log n \leq n^2 \quad \forall n \geq n_0$

$n_0 = 1$

• $3n^3 + 4 \in \Omega(n^2)$

$c = 3$ $\quad 0 \leq 3 \cdot n^2 \leq 3n^3 \leq 3n^3 + 4 \quad \forall n \geq n_0 = 1$

$n_0 = 1$

III

a) $T(n) = 256\,T\left(\frac{n}{4}\right) + n^3$

$= \cancel{256} \sum_{i=1}^{256} T\left(\frac{1}{4}\cdot n\right) + \Theta(n^3)$

Also: $\alpha_i = \frac{1}{4}$, $i=1,\dots,256$

$m = 256$

$k = 3$

$$\sum_{i=1}^{m} \alpha_i^k = \sum_{i=1}^{256} \left(\frac{1}{4}\right)^3 = 256 \cdot \frac{1}{64} > 1$$

$\hookrightarrow$ 3. Fall, wir suchen also c mit $\sum_{i=1}^{m} \alpha_i^c = 1$

$$\sum_{i=1}^{256} \left(\frac{1}{4}\right)^c = 1$$

$$\Leftrightarrow \quad 256 \left(\frac{1}{4}\right)^c = 1$$

$$\Leftrightarrow \quad \left(\frac{1}{4}\right)^c = \frac{1}{256}$$

$$\Leftrightarrow \quad 4^c = 256$$

$$\Leftrightarrow \quad c = \log_4 256$$

$$\Leftrightarrow \quad c = 4$$

$$\Rightarrow \quad T(n) \in \Theta(n^c) = \Theta(n^4)$$

$\overline{\text{IV}}$

b) $T(n) = 27 \cdot T\left(\frac{n}{3}\right) + n^3$

$\alpha_i = \cancel{27}\,\frac{1}{3}, \quad i = 1, \dots, 27$

$m = 27$

$k = 3$

$$\sum_{i=1}^{m} \alpha_i^{k} = \sum_{i=1}^{27} \left(\frac{1}{3}\right)^3 = 27 \cdot \frac{1}{27} = 1 \qquad \rightarrow \text{Fall 2}$$

$$\Rightarrow T(n) \in \Theta(n^{k} \log n) = \Theta(n^3 \log n)$$

c) $T(n) = 3\, T\left(\frac{n}{4}\right) + n^2$

$\alpha_i = \frac{1}{4}, \quad i = 1, \dots, 3$

$m = 3$

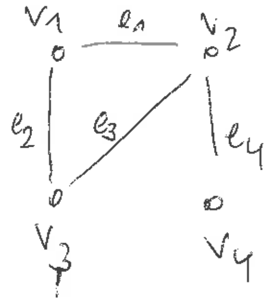
$k = 2$

$$\sum_{i=1}^{m} \alpha_i^{k} = \sum_{i=1}^{3} \left(\frac{1}{4}\right)^2 = 3 \cdot \frac{1}{16} < 1 \qquad \rightarrow \text{Fall 1}$$

$$\Rightarrow T(n) \in \Theta(n^{k}) = \Theta(n^2)$$

(V) ~~(III)~~

(1) adjacency matrix



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

(2) incidence matrix

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

(3) adjaceny list:

v_1: v_2, v_3

v_2: v_1, v_4

v_3: v_1, v_2

v_4: v_2

Proof of Theorem 5.2

Let v be a vertex with odd degree $\delta(v)$.

$\Rightarrow$ # of edges in an Eulerian path P leading to v

$\neq$ # of edges in P leaving v

$\Rightarrow$ P must start or end in v.

Thus:

(1) For an Eulerian path we have one start and one end vertex

(2) For an Eulerian walk: start vertex = end vertex w

$\hookrightarrow$ has equal # of edges entering and leaving

$\Rightarrow \delta(w)$ even

$\square$

Proof of Theorem 5.4

HW: 1.1(a) $\quad \sum_{i=1}^{n} \delta(v_i) = 2m$ is even

$\Rightarrow$ # of vertices with odd degree is even.

$\square$

Proof of Theorem 5.15

Let $W = s, e_1, v_1, \dots, v_m, e_{m+1}, t$ be a walk from s to t.

idea:

delete cycles on the way

Assume no path exists.

Consider a walk with the minimum number of vertices visited twice. Let v_i be one of these.

II

$W' = s, e_1, v_1, \dots, v_i, e_{i+1}, \dots, v_i, e_k, \dots, t$

first visit — second visit

$\Rightarrow W = s, e_1, v_1, \dots, v_i, e_k, \dots, t$ is a shorter walk ↯ □

DFS in a maze: A1

(a) Proof: Consider an arbitrary edge $e = (u,v)$, used during DFS. u, v were included in Q and R.

~~Q~~ is

Contructed graph is a tree (i.e. ∃ unique path from all vertices considered after v to v)
+ DFS is finite

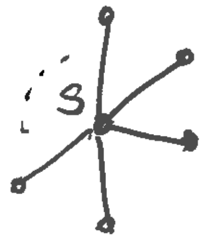
⇒ After a finite # of steps the edge is used again from v to u.

⇒ edge used twice.

To use e a third time, v would have to be selected again.
As the next vertex is chosen from $V \setminus R$ (and v is in R) this is impossible.

(b) Proof: Graph with $n+1$ vertices
⇒ spanning tree, which DFS constructs, has n edges
If each edge is used twice: $2n$ steps
BUT: exit found ⇒ last edge not used on the way back
⇒ $2n-1$ steps

(c) Proof:



graph with node s and edges between s and all other nodes ≙ star.

The node the strategy visits last is the exit.

III

Proof Theorem 5.25

We delete parallel edges – only the cheapest is left in the graph, the others are redundant.

$\Rightarrow m = O(n^2)$

For ③: data structure that manages connected components of T.

Test in ③ whether $T + e_i = \{v, w\}$ results in cycle $\iff$ v, w are in the same connected component (CC) (edge connects two vertices in the same connected component)

* For each component we keep a directed tree (an arborescence!) with a unique root; each vertex gets a unique predecessor:

component
directed tree of logarithmic height

B with $V(B) = V(T)$
and $|E(B)| = |E(T)|$

apiece: CC of B induced by the same set of vertices as the CC of T

* When we test $e_i = \{v, w\}$ in 3., we find the roots r_v and r_w of the CC in B that contain v and w, respectively.

Time? proportional to the length of an r_v-v-path in B
+ — " — r_w-w-path — " —

$\rightarrow$ $O(\log n)$ (= height of the tree)
$\hookrightarrow$ We still need to show that!

* Test $r_v = r_w$
  - If yes: test next edge
  - If no: we add e_i to T_1 and we need to add an edge to B.
    Let $h(r)$ be the maximum length of a path from r in B.
    - If $h(r_v) \geq h(r_w)$: add the edge (r_v, r_w) to B
    - otherwise: — " — (r_w, r_v) to B

  Change of $h(r_v)$?
  * $h(r_v) = h(r_w) \Rightarrow$ increases $h(r_v)$ by one
  * otherwise: the new root has the same h-value as before

Claim 5.26: A directed subtree of B with root r contains at least $2^{h(r)}$ vertices.

Proof by induction:
* we start with: $B = (V(G), \emptyset)$, $h(v) = 0$, claim holds ✓
* to show: property holds true when we add an edge (x, y) to B.
  - if $h(r)$ does not change – clear
  - otherwise: before we had $h(x) = h(y)$
    $\hookrightarrow$ Both CC have at least $2^{h(x)}$ vertices
    $\Rightarrow$ New CC, rooted in x, has at least $2 \cdot 2^{h(x)} = 2^{h(x)+1}$ vertices

✓

- ⇨ h(r) <= log n
- ⇨ logarithmic height
- ⇨ complexity of O(m log n) □