III

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

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

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

$m = 256$

$k = 3$

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

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

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

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

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

$$\Leftrightarrow 4^c = 256$$

$$\Leftrightarrow c = \log_4 256$$

$$\Leftrightarrow c = 4$$

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

IV

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

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

$m = 27$

$k = 3$

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

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

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

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

$m = 3$

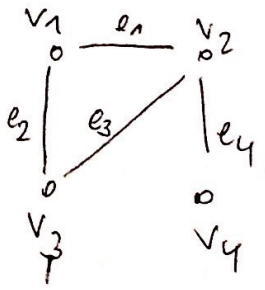
$k = 2$

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

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

(V)

(1) adjacency matrix



	v_1	v_2	v_3	v_4
v_1	0	1	1	0
v_2	1	0	1	1
v_3	1	1	0	0
v_4	0	1	0	0

(2) incidence matrix

	e_1	e_2	e_3	e_4
v_1	1	1	0	0
v_2	1	0	1	1
v_3	0	1	1	0
v_4	0	0	0	1

(3) adjacency 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

↳ has equal # of edges entering and leaving

$\Rightarrow$ $\delta(w)$ even $\square$

Proof of Theorem 5.4

~~#~~ HW: 1.1(a) $\sum_{i=1}^{n} \delta(v_i) = 2m$ 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.

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

(v_i: 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.

~~Gras~~

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

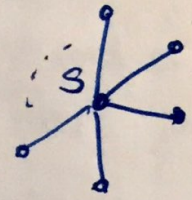
$\Rightarrow$ After a finite # of steps the edge is used again from v to u.

$\Rightarrow$ 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.

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

Proof Theorem 5.25

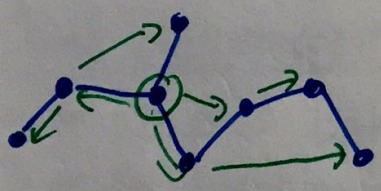
We delete parallel edges - only the cheapest is left in the graph, the others are redundant.

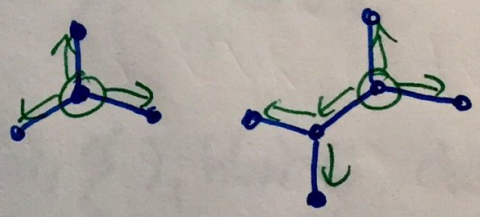
$\Rightarrow m = O(n^2)$

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

Test in (3) whether $T + e_i = \{v,w\}$ results in cycle $\Longleftrightarrow$ 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 $\mathcal{B}$ that contain v and w, respectively.
  Time? proportional to the length of an r_v-v-path in $\mathcal{B}$
  + —"— r_w-w-path —"—
  → $O(\log n)$ (= height of the tree)
  ↳ 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 $\mathcal{B}$.
    Let $h(r)$ be the maximum length of a path from r in $\mathcal{B}$.
    • If $h(r_v) \geq h(r_w)$: add the edge (r_v, r_w) to $\mathcal{B}$
    • otherwise: —"— (r_w, r_v) to $\mathcal{B}$

  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 $\mathcal{B}$ with root r contains at least $2^{h(r)}$ vertices.

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

$\Rightarrow h(r) \in \log n$

$\Rightarrow$ logarithmic height

$\Rightarrow$ complexity of $O(m \log n)$

□