

AuD

große Übung: 20.11.2014

Christian Scheffer

Hunger?

Hunger?

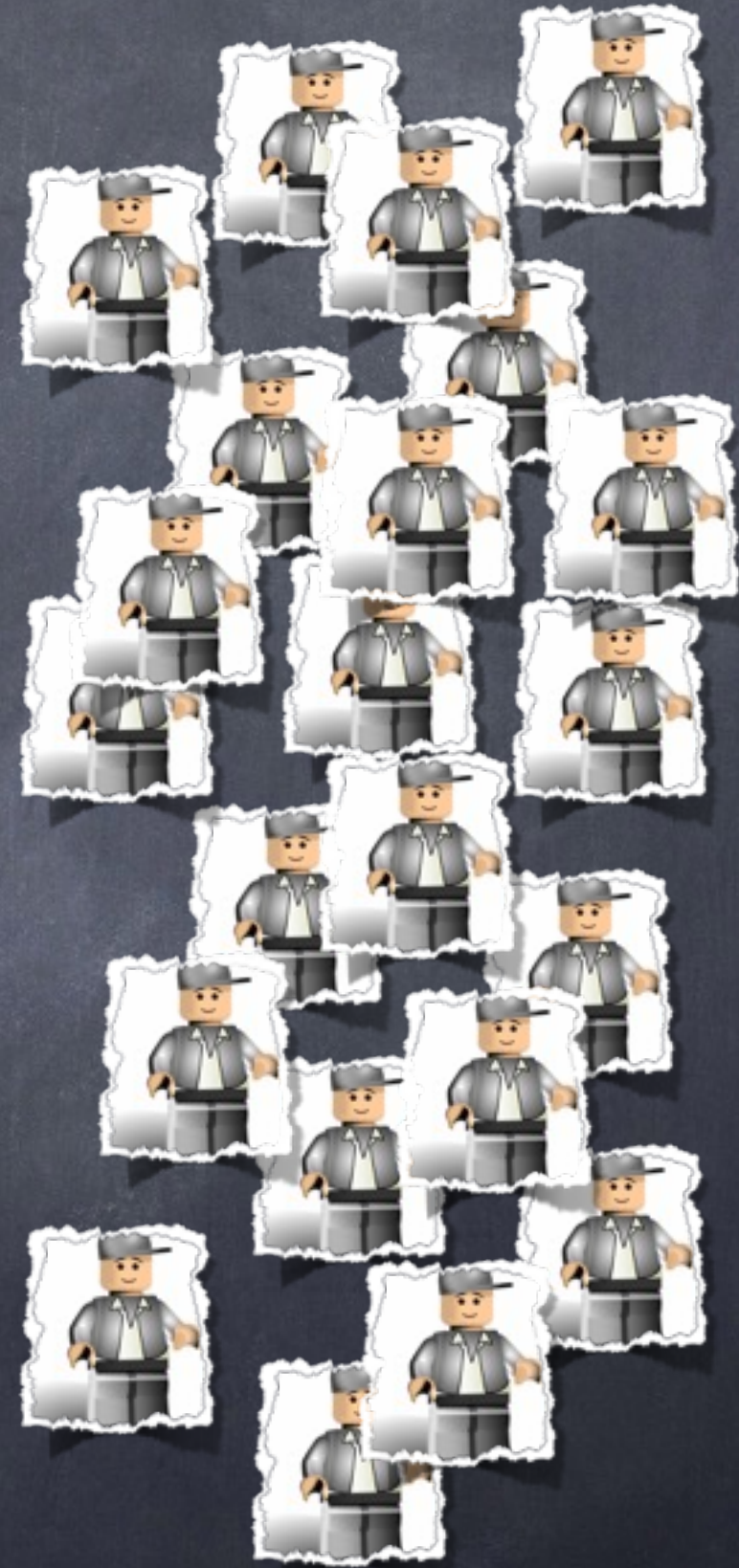




Hunger?



Hunger?



Hunger?



Hunger?



Hunger?



70



70



70



70



70



70



70



70



70



70



70



70



70



70



70



69



70



70



70



70



69



70



70



70



70



69



70



70



70



70



69



70



70



69



70



69



70



70



69



70



69



70



70



69



70



69



69



70



69



70



69



69



70



69



70



69



69



70



69



70



69



69



70



69



70



69



68



70



69



70



69



68



70



69



70



69



68



70



69



70



69



67



70



69



70



1



0



0



0



10



1



0



0



0



10



1



0



0



0



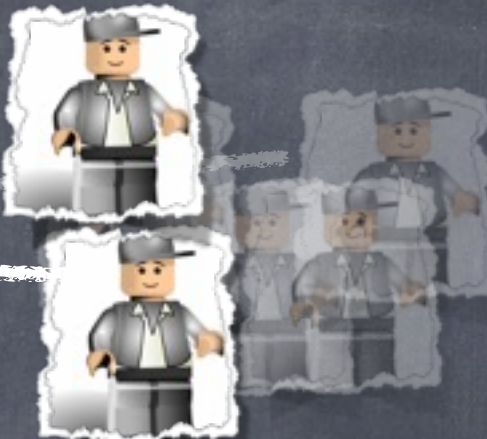
10



1



0



0



0



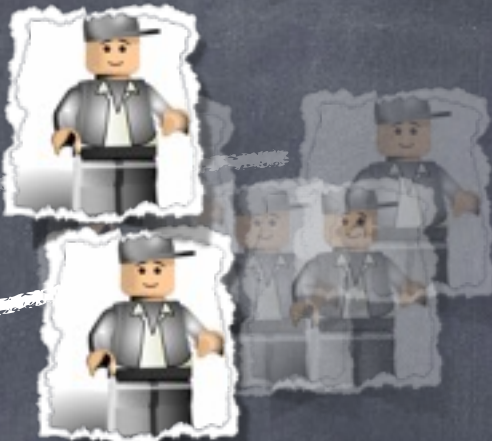
10



1



0



0



0



10



1



0



0



0



10



1



0



0



0



10



1



0



0



0



10



1



0



0



0



10



1



0



0



0



10



1



0



0



0



9



1



0



0



0



9



1



0



0



0



9



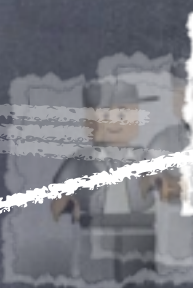
1



0



0



0



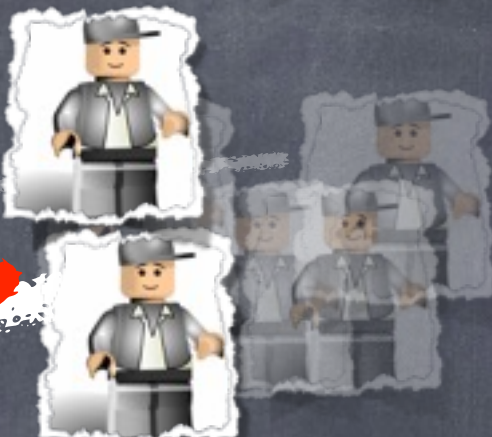
9



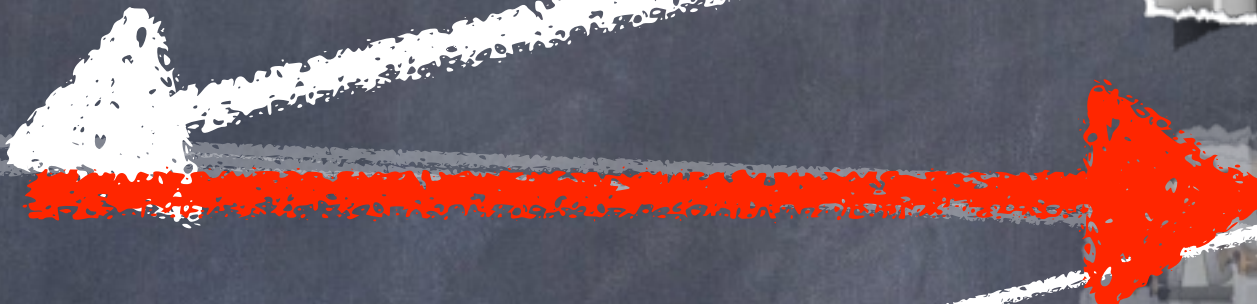
1



0



0



0



9



1



0



0



0



9



1



0



0



0



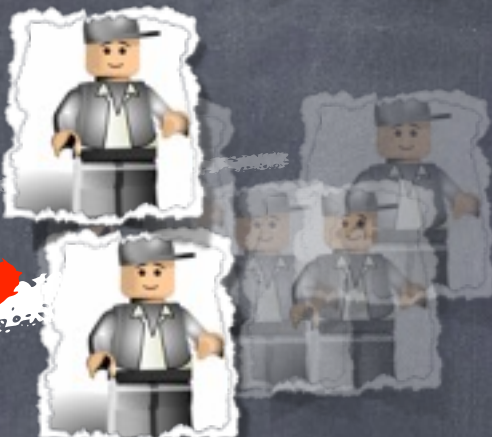
9



1



0



0



0



9



1



gerichtete Graphen

0



0



0



9



Pizzabestellung



⋮



V_2

Pizzabestellung



⋮

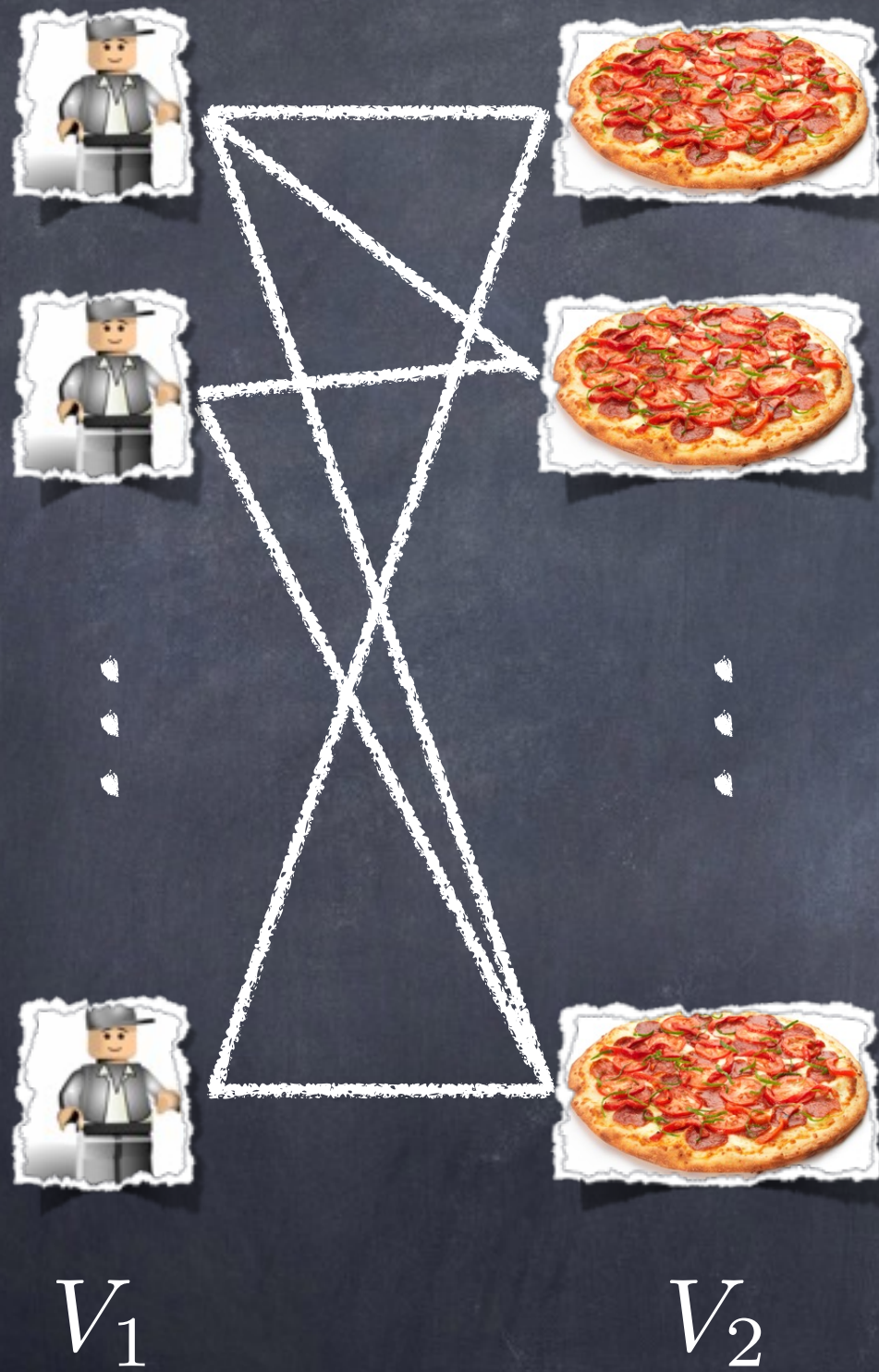
⋮



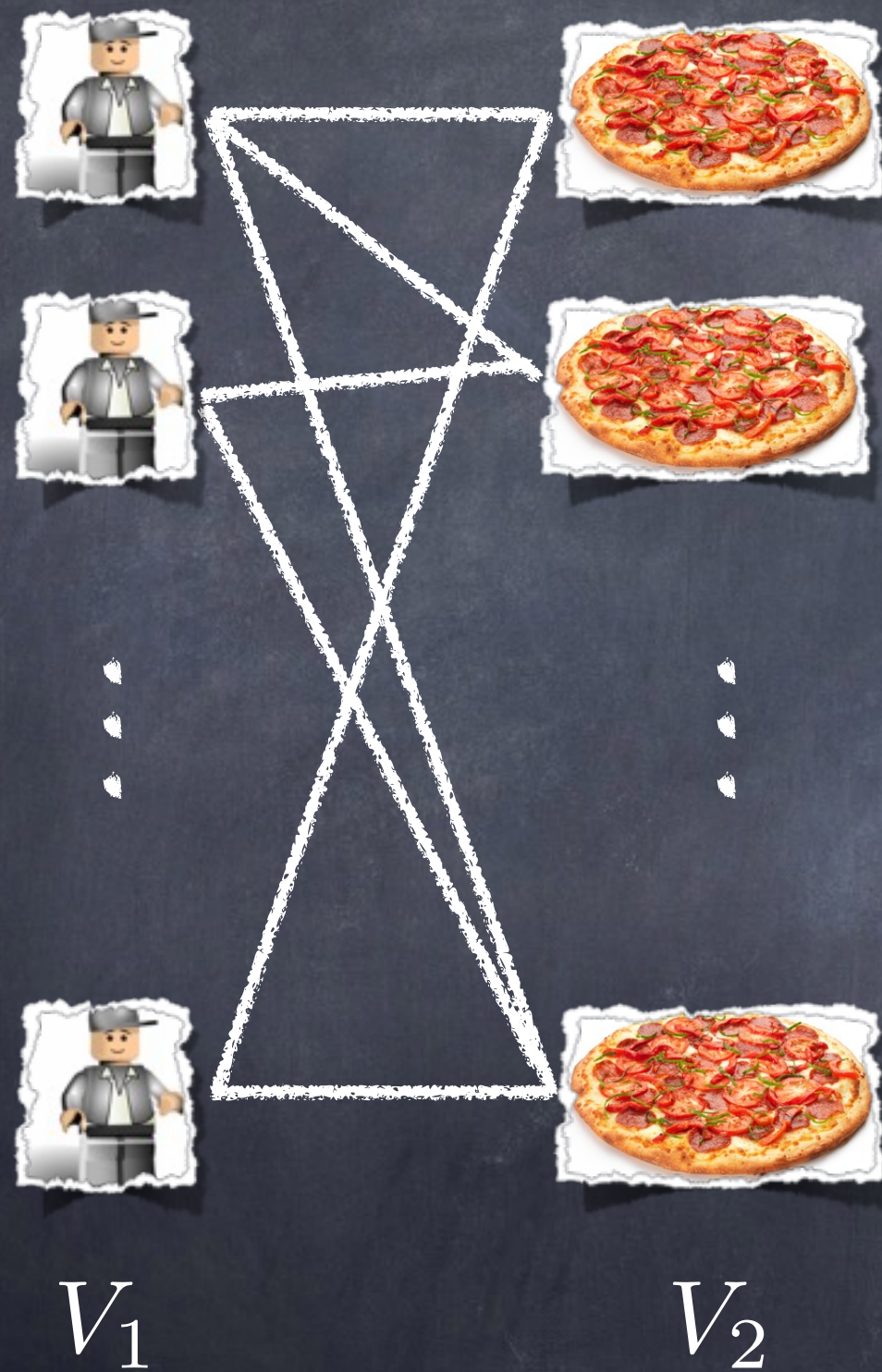
V_1

V_2

Pizzabestellung

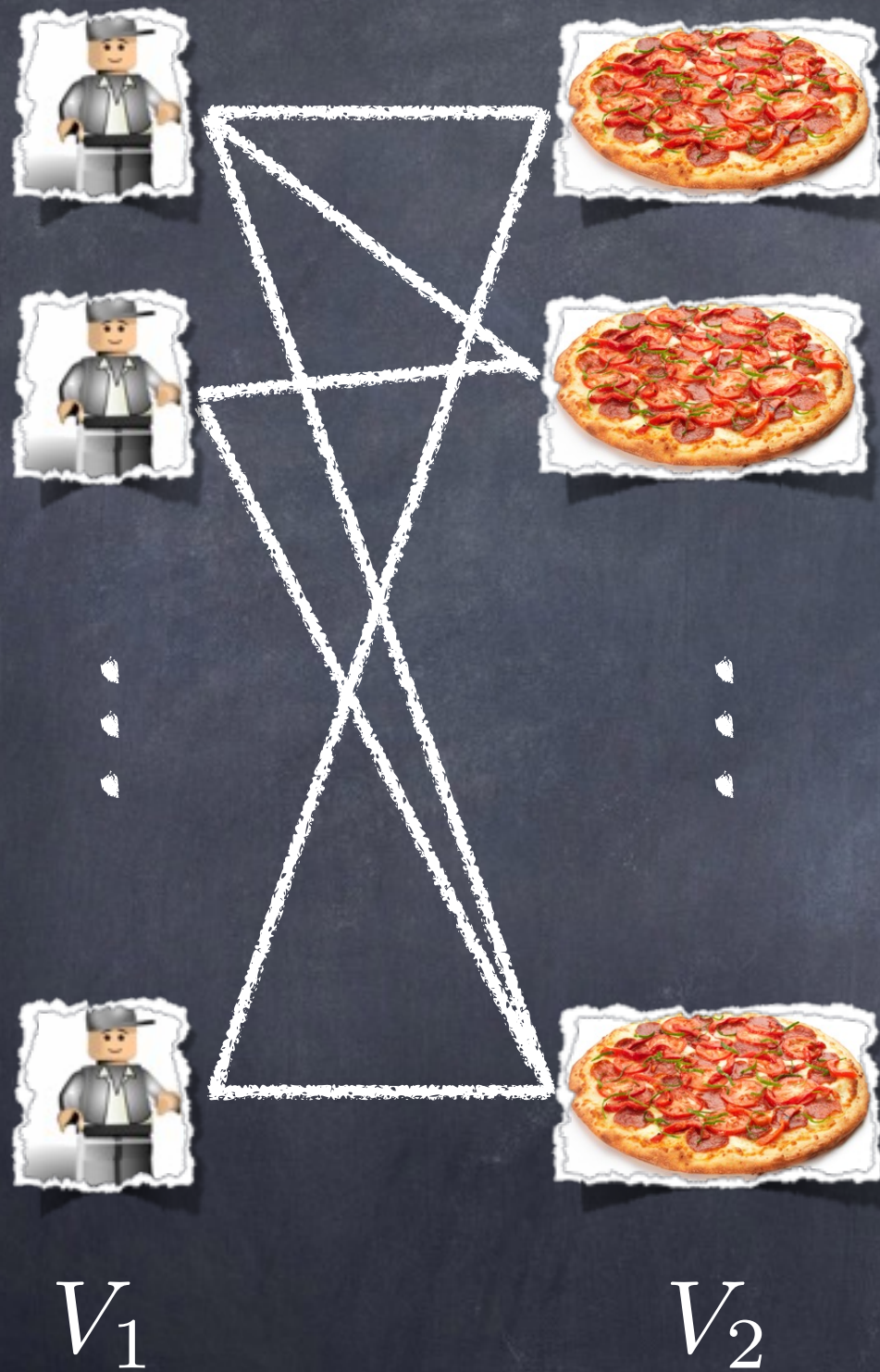


Pizzabestellung



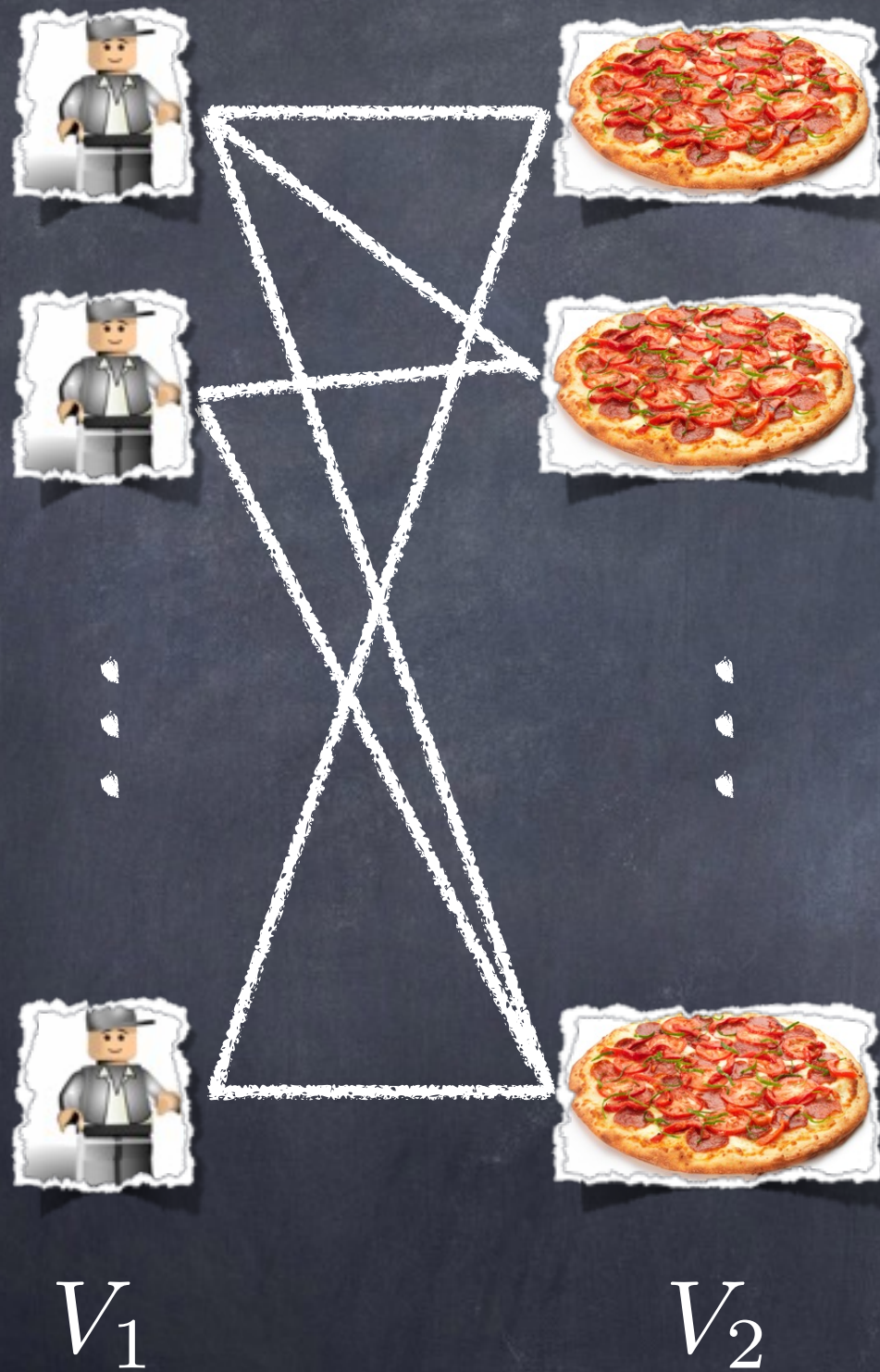
$$G := (V, E)$$

Pizzabestellung



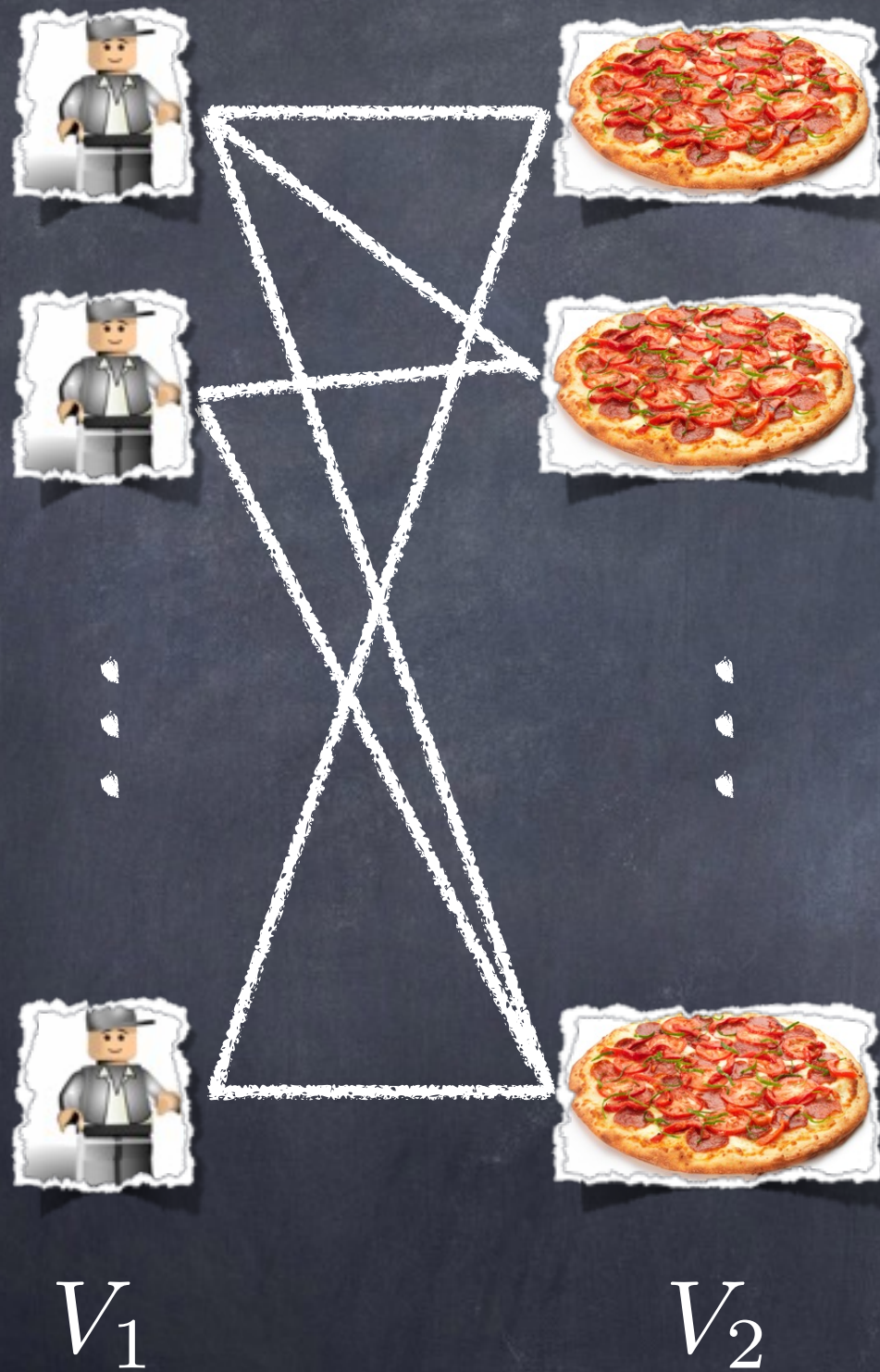
- $G := (V, E)$
- $V := V_1 \cup V_2$

Pizzabestellung



- $G := (V, E)$
- $V := V_1 \cup V_2$
- $E \subset V_1 \times V_2$

Pizzabestellung



- $G := (V, E)$

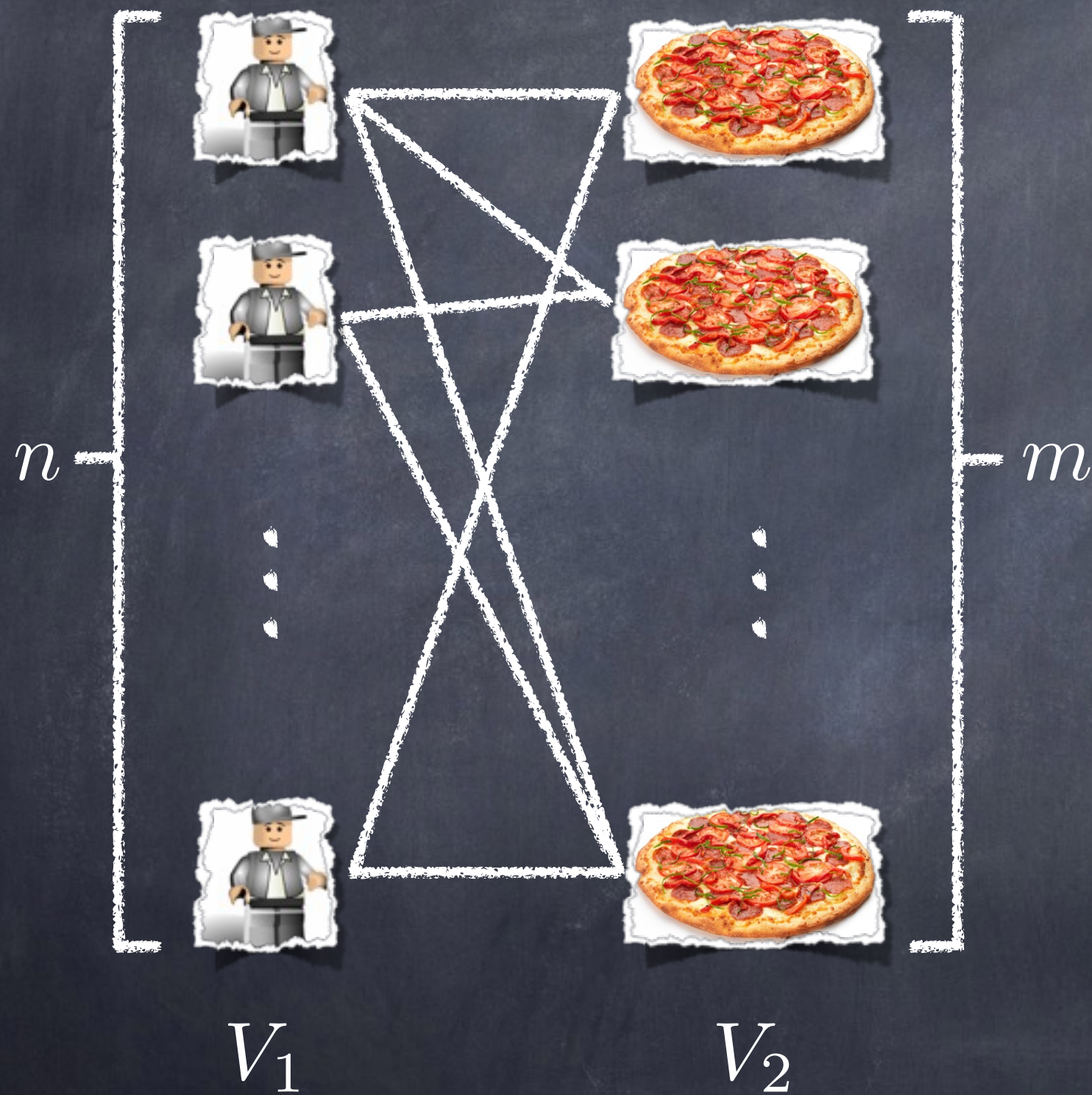
- $V := V_1 \cup V_2$

- $E \subset V_1 \times V_2$

mit $(v, w) \in E$

$:\Leftrightarrow$ Student v mag
Pizza w

Pizzabestellung



- $G := (V, E)$

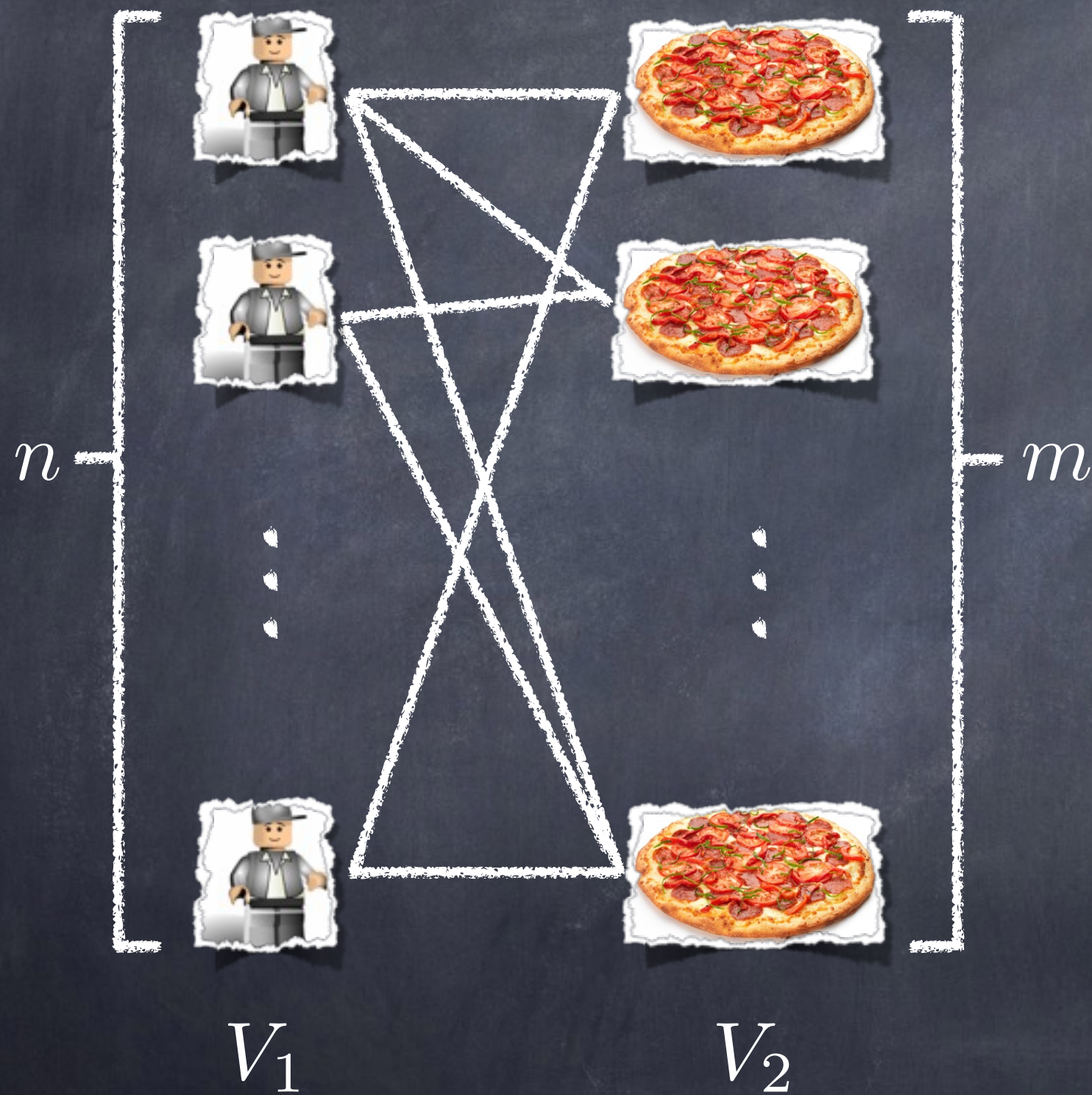
- $V := V_1 \cup V_2$

- $E \subset V_1 \times V_2$

mit $(v, w) \in E$

$:\Leftrightarrow$ Student v mag
Pizza w

Pizzabestellung



- $G := (V, E)$

- $V := V_1 \cup V_2$

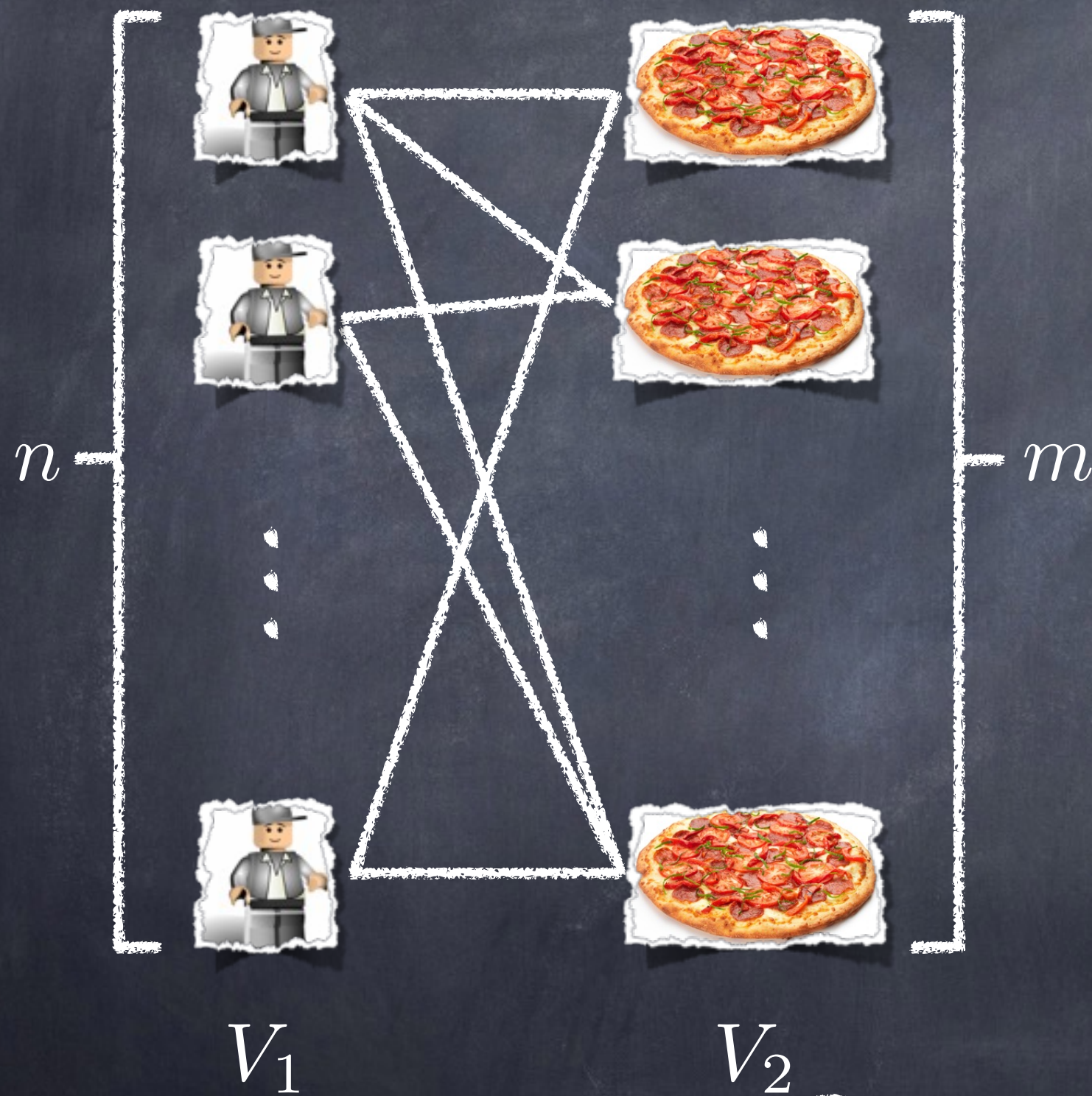
- $E \subset V_1 \times V_2$

mit $(v, w) \in E$

$:\Leftrightarrow$ Student v mag
Pizza w

- $\forall v \in V_1 : \delta(v) \leq m$

Pizzabestellung



- $G := (V, E)$

- $V := V_1 \cup V_2$

- $E \subset V_1 \times V_2$

mit $(v, w) \in E$

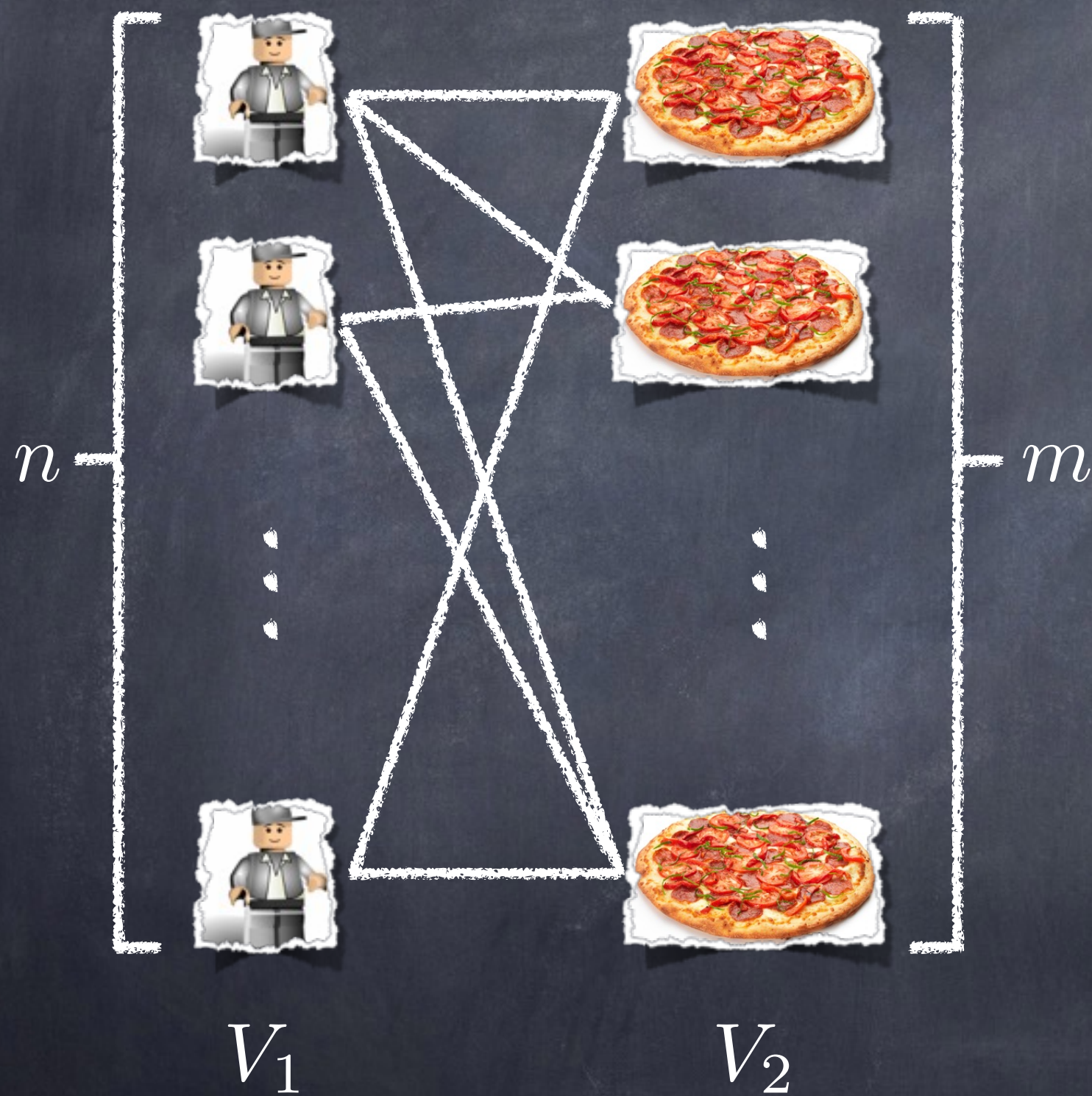
\Leftrightarrow Student v mag
Pizza w

- $\forall v \in V_1 : \delta(v) \leq m$

- $\forall w \in V_2 : \delta(w) \leq 30$

Pizza hat nur 30 Stücke

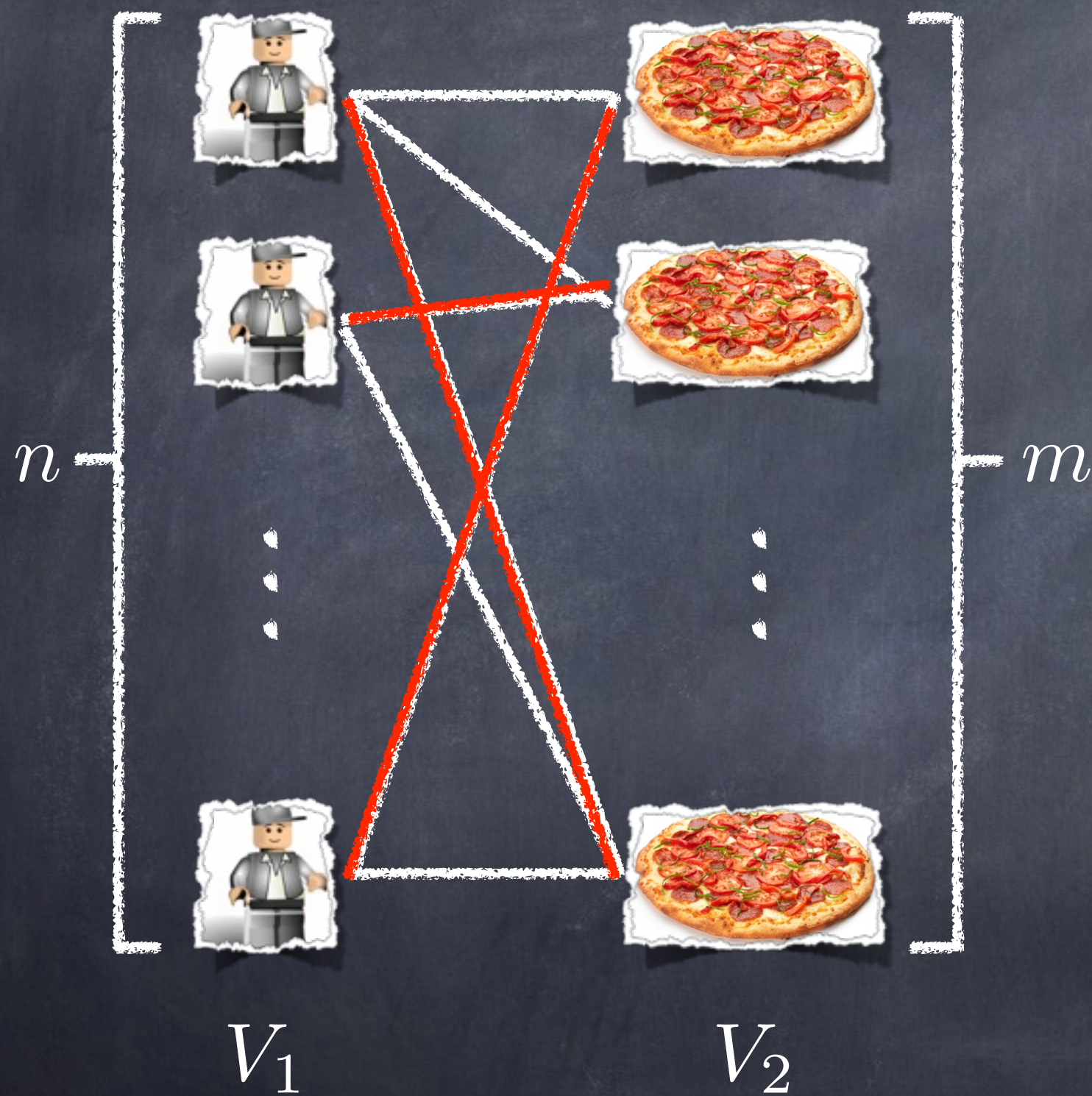
Pizzabestellung



- $G := (V, E)$
- $V := V_1 \cup V_2$

mögl. Einteilung

Pizzabestellung

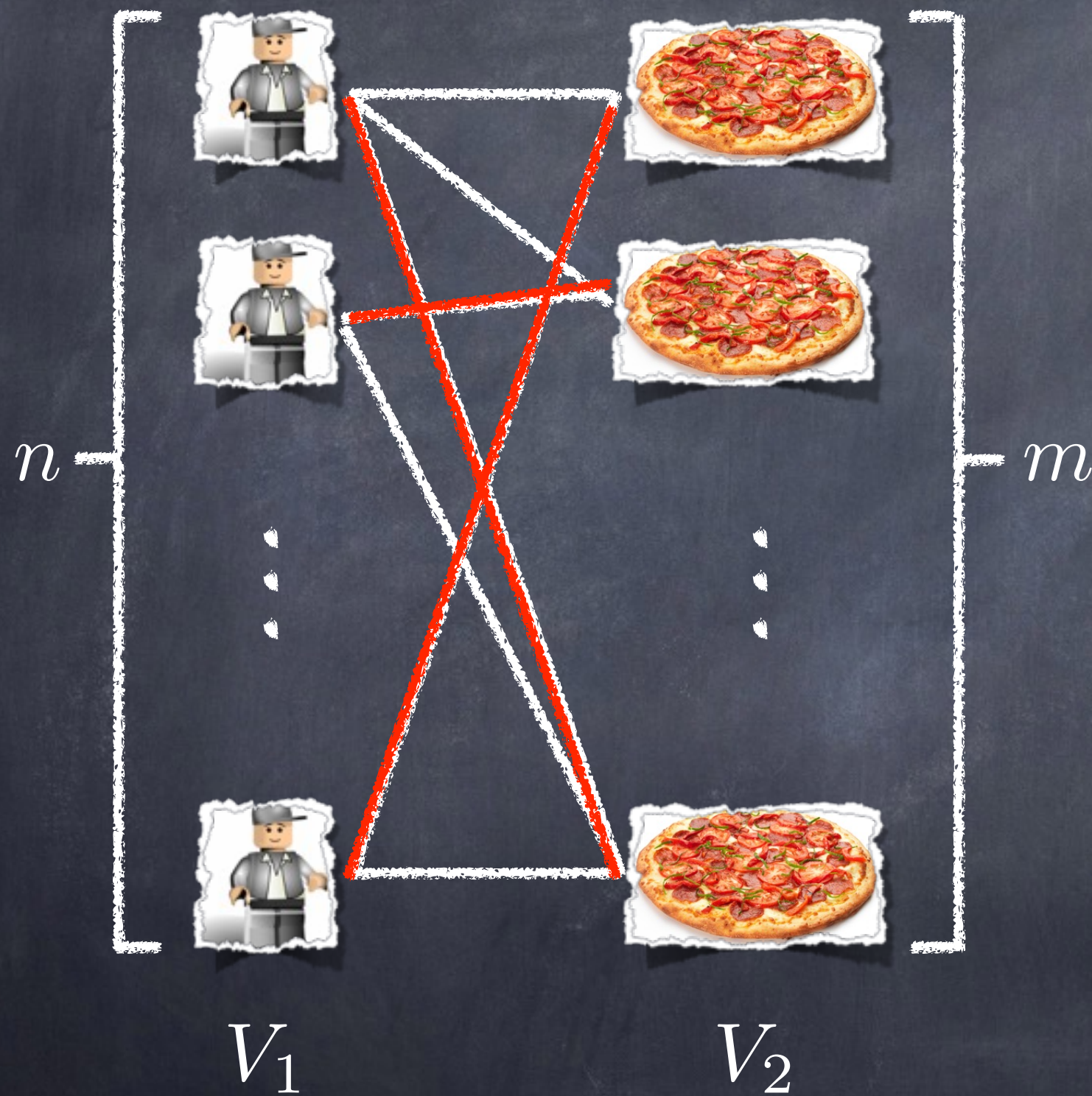


- $G := (V, E)$

- $V := V_1 \cup V_2$

mögl. Einteilung

Pizzabestellung



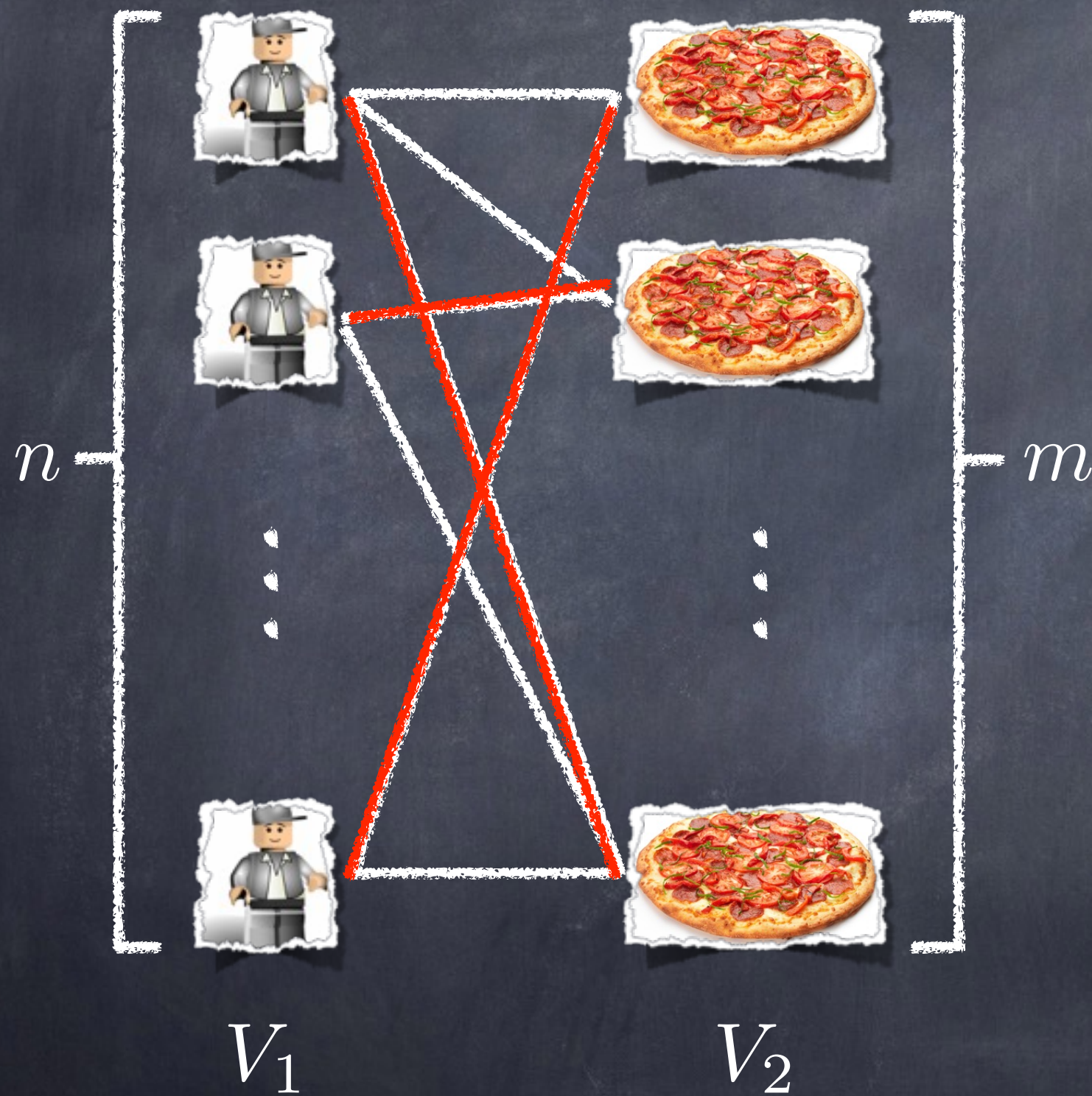
- $G := (V, E)$
- $V := V_1 \cup V_2$

mögl. Einteilung

- $G' := (V, E')$

mit $E' \subseteq E$ und
 $\forall v \in V_1 : \delta(v) \leq 1$

Pizzabestellung



- $G := (V, E)$
- $V := V_1 \cup V_2$

mögl. Einteilung

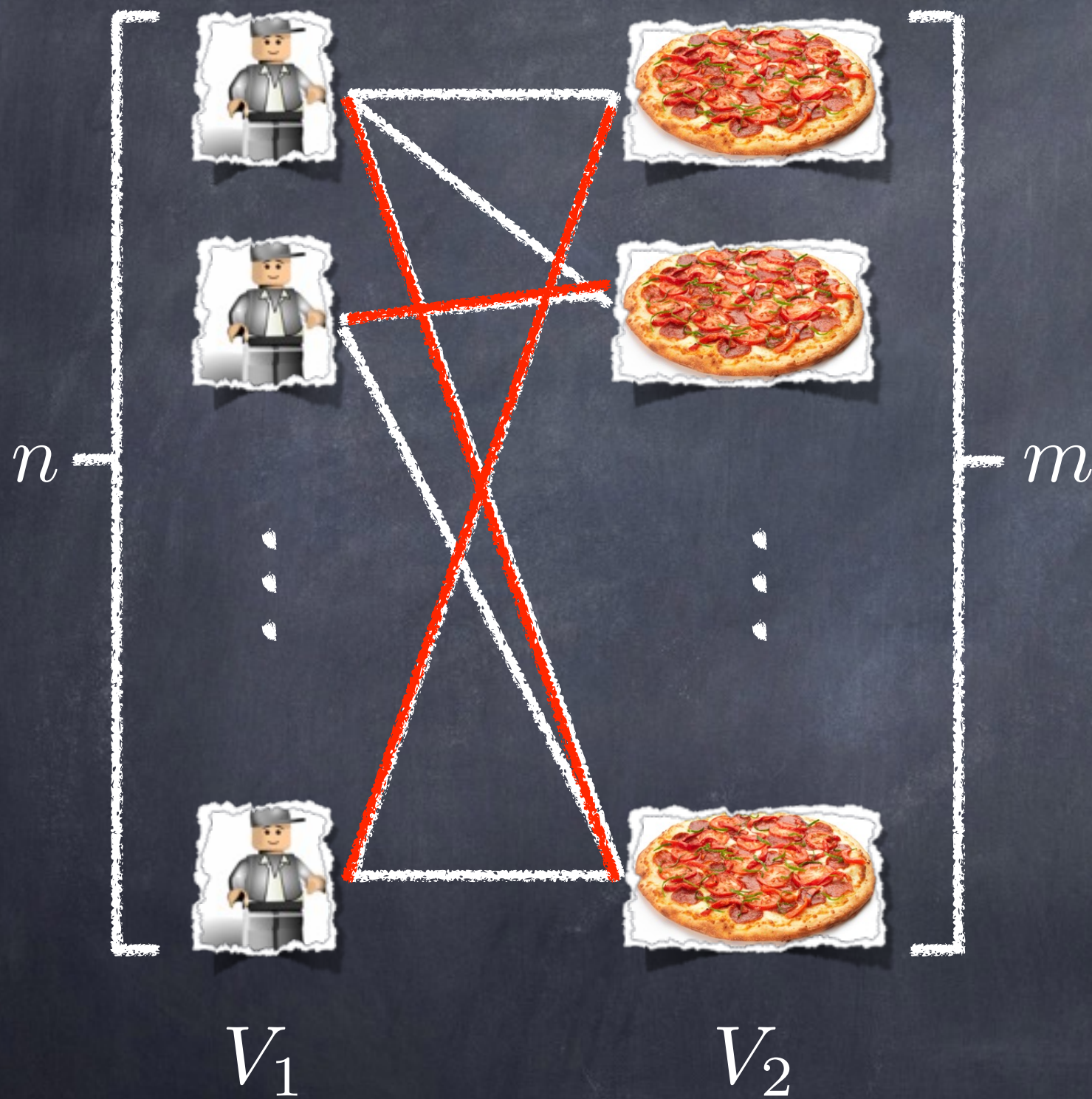
- $G' := (V, E')$

mit $E' \subseteq E$ und

$$\forall v \in V_1 : \delta(v) \leq 1$$

- $val(G') := |E'|$

Pizzabestellung



- $G := (V, E)$
- $V := V_1 \cup V_2$

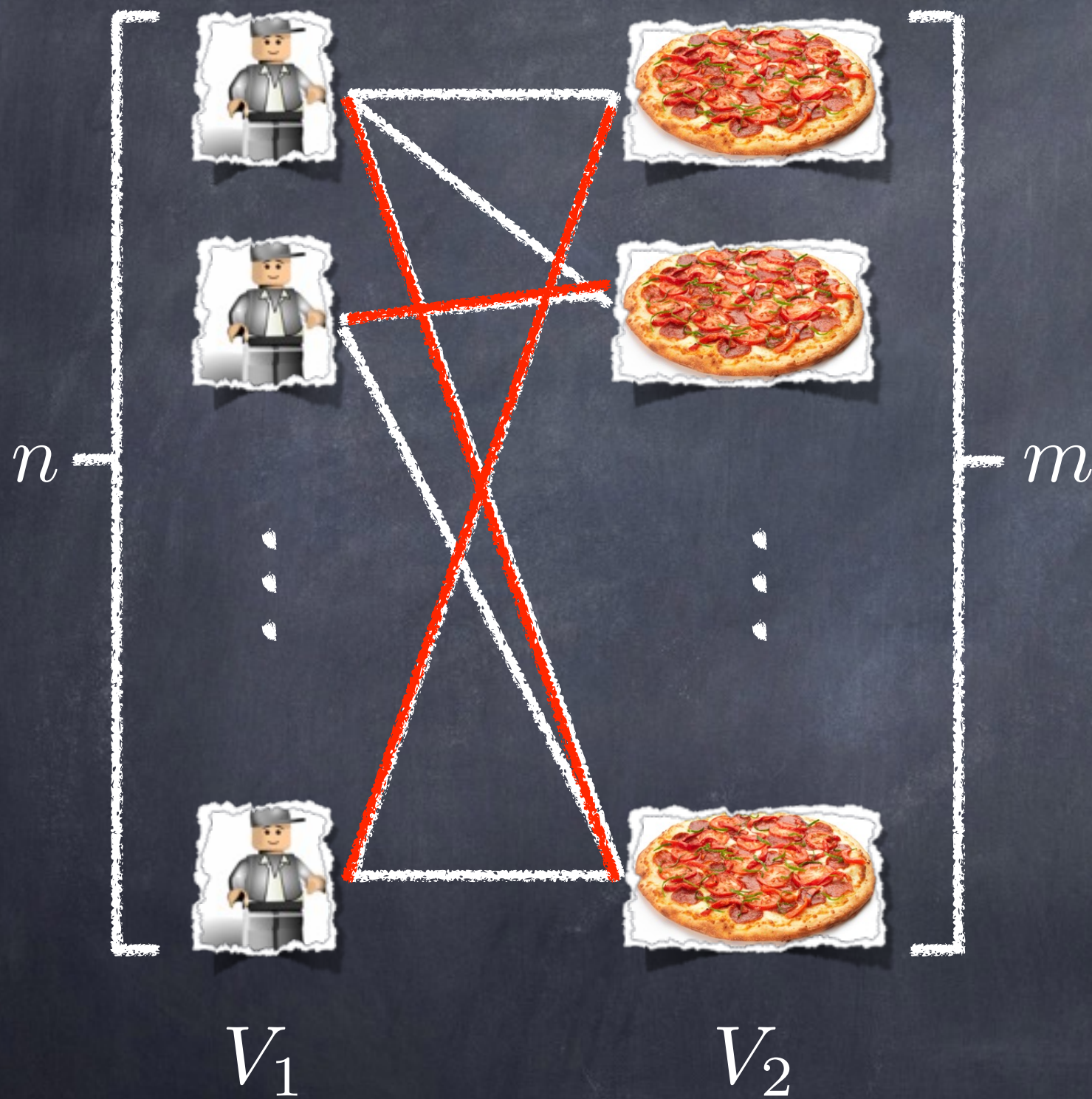
mögl. Einteilung

- $G' := (V, E')$

mit $E' \subseteq E$ und
 $\forall v \in V_1 : \delta(v) \leq 1$

- $val(G') := |E'|$
- $\arg \max_{G' \subseteq G} val(G')$

Pizzabestellung



- $G := (V, E)$
- $V := V_1 \cup V_2$

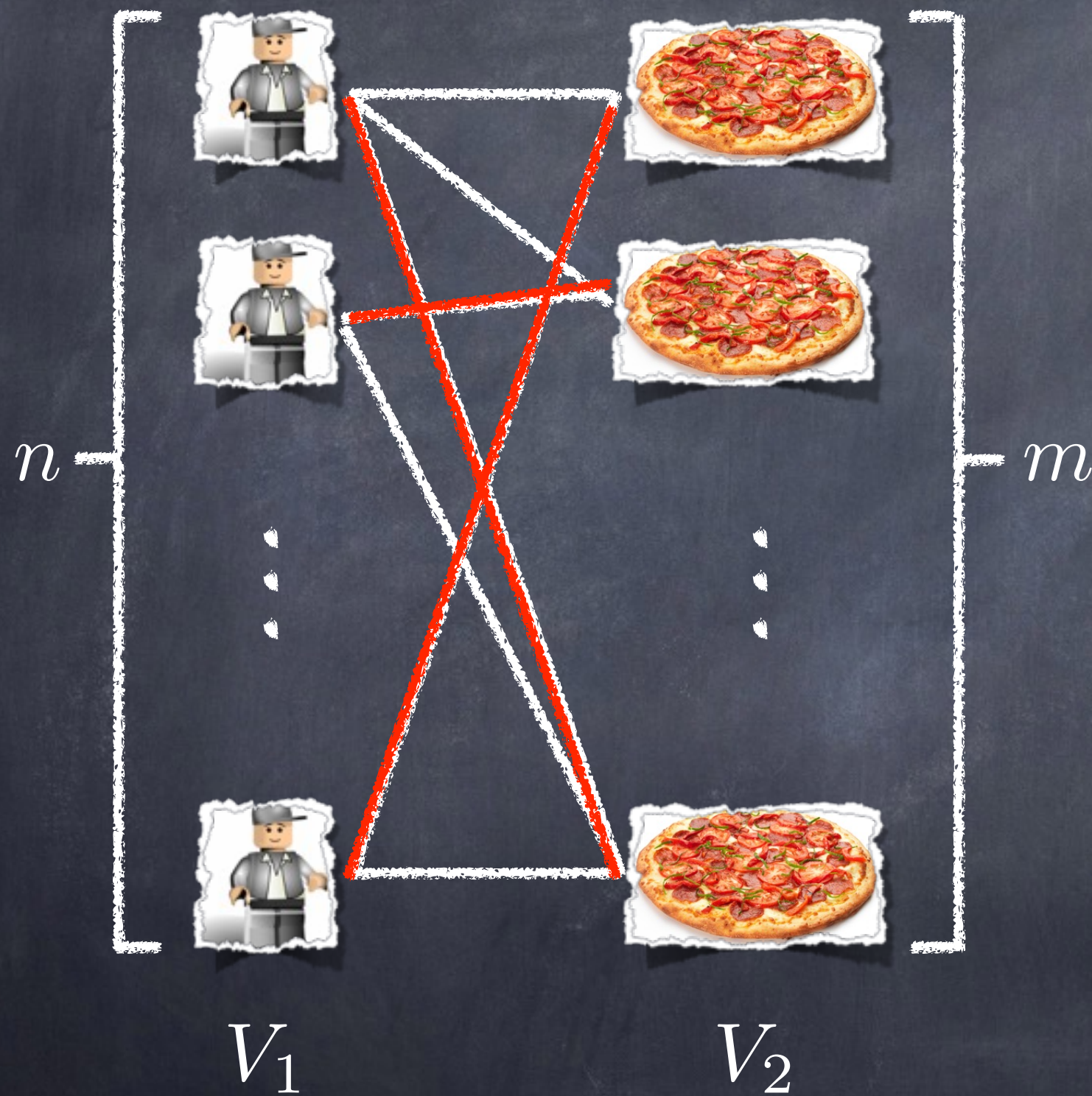
mögl. Einteilung

- $G' := (V, E')$

mit $E' \subseteq E$ und
 $\forall v \in V_1 : \delta(v) \leq 1$

- $val(G') := |E'|$
- $\arg \max_{G' \subseteq G} val(G')$
- $|E'| \leq n$

Pizzabestellung



- $G := (V, E)$
- $V := V_1 \cup V_2$

mögl. Einteilung

- $G' := (V, E')$

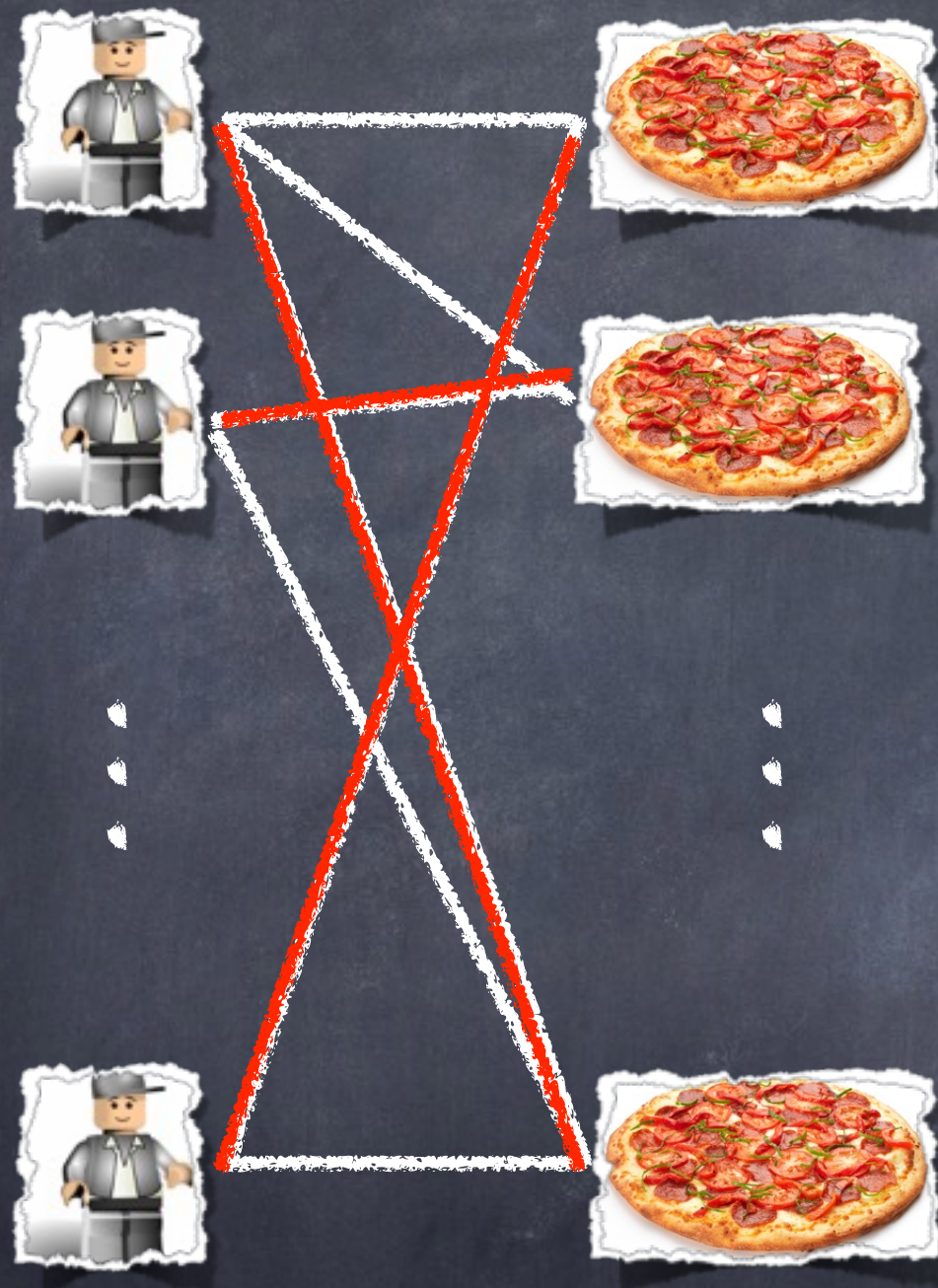
mit $E' \subseteq E$ und
 $\forall v \in V_1 : \delta(v) \leq 1$

- $val(G') := |E'|$
- $\arg \max_{G' \subseteq G} val(G')$

- $|E'| \leq n$

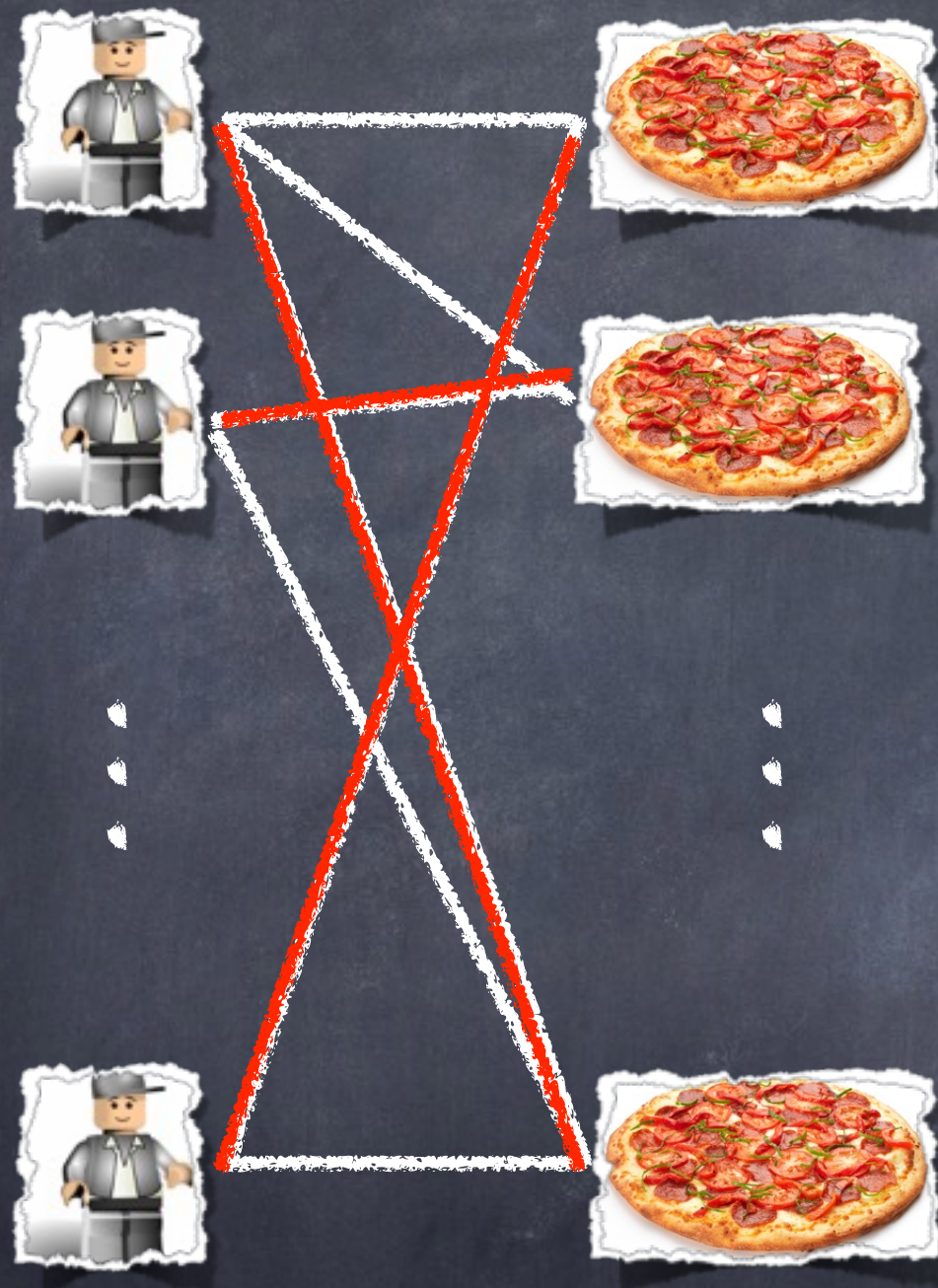
- $|E'| \leq 30 \cdot m$

Pizzabestellung



Pizza(G)

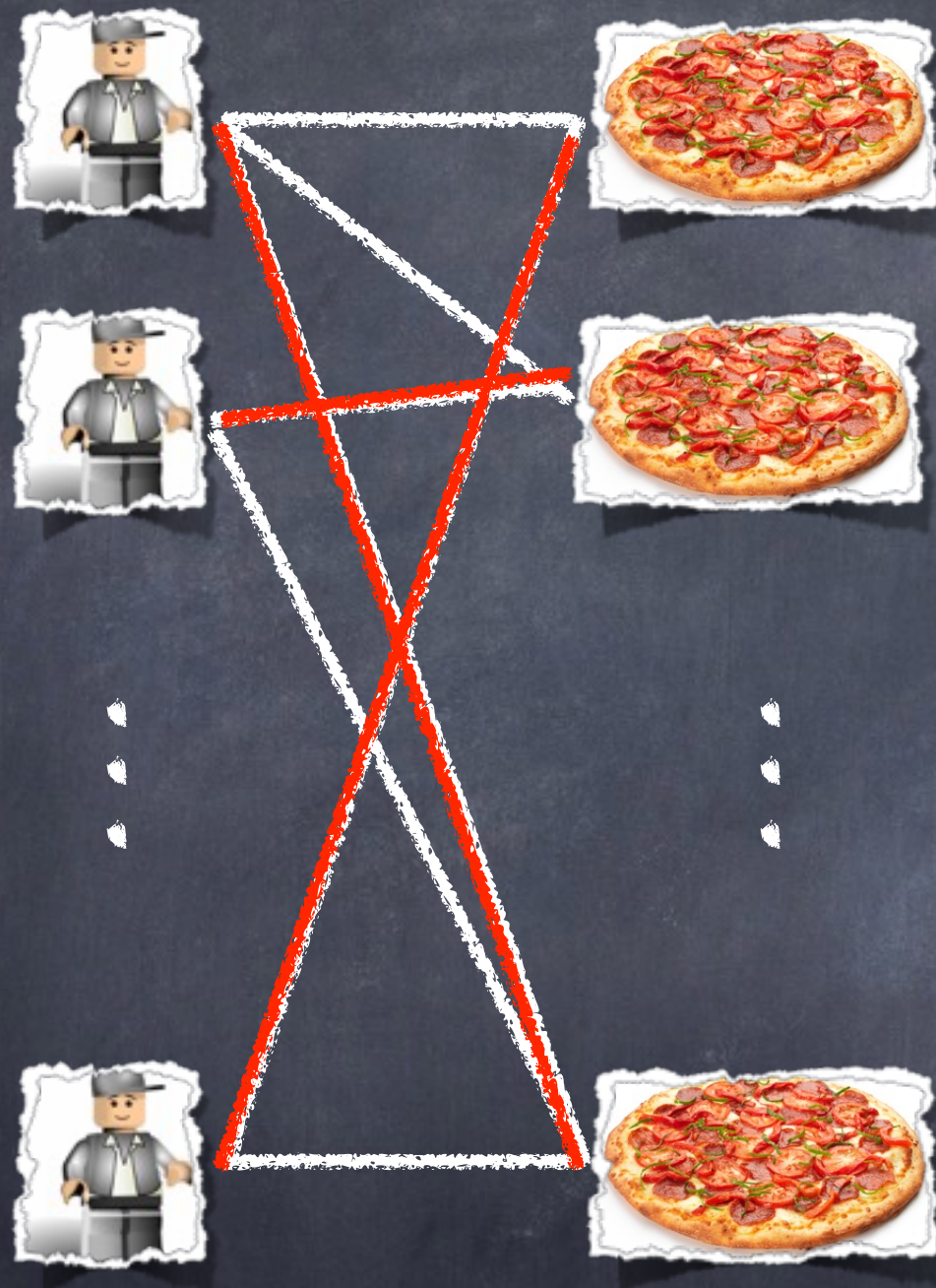
Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

Pizzabestellung

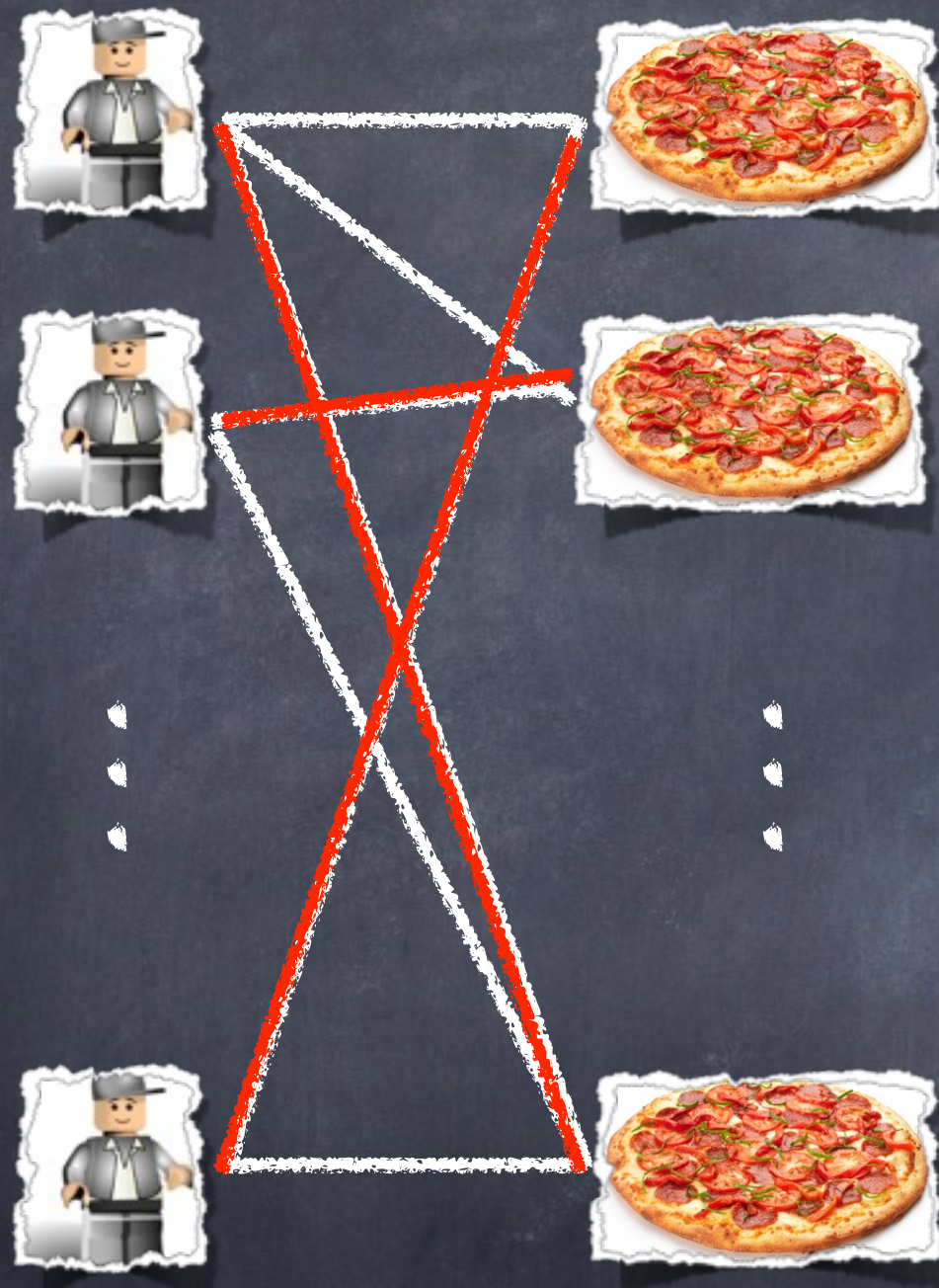


Pizza(G)

$$E' := \emptyset$$

$$G' := (V, E')$$

Pizzabestellung



Pizza(G)

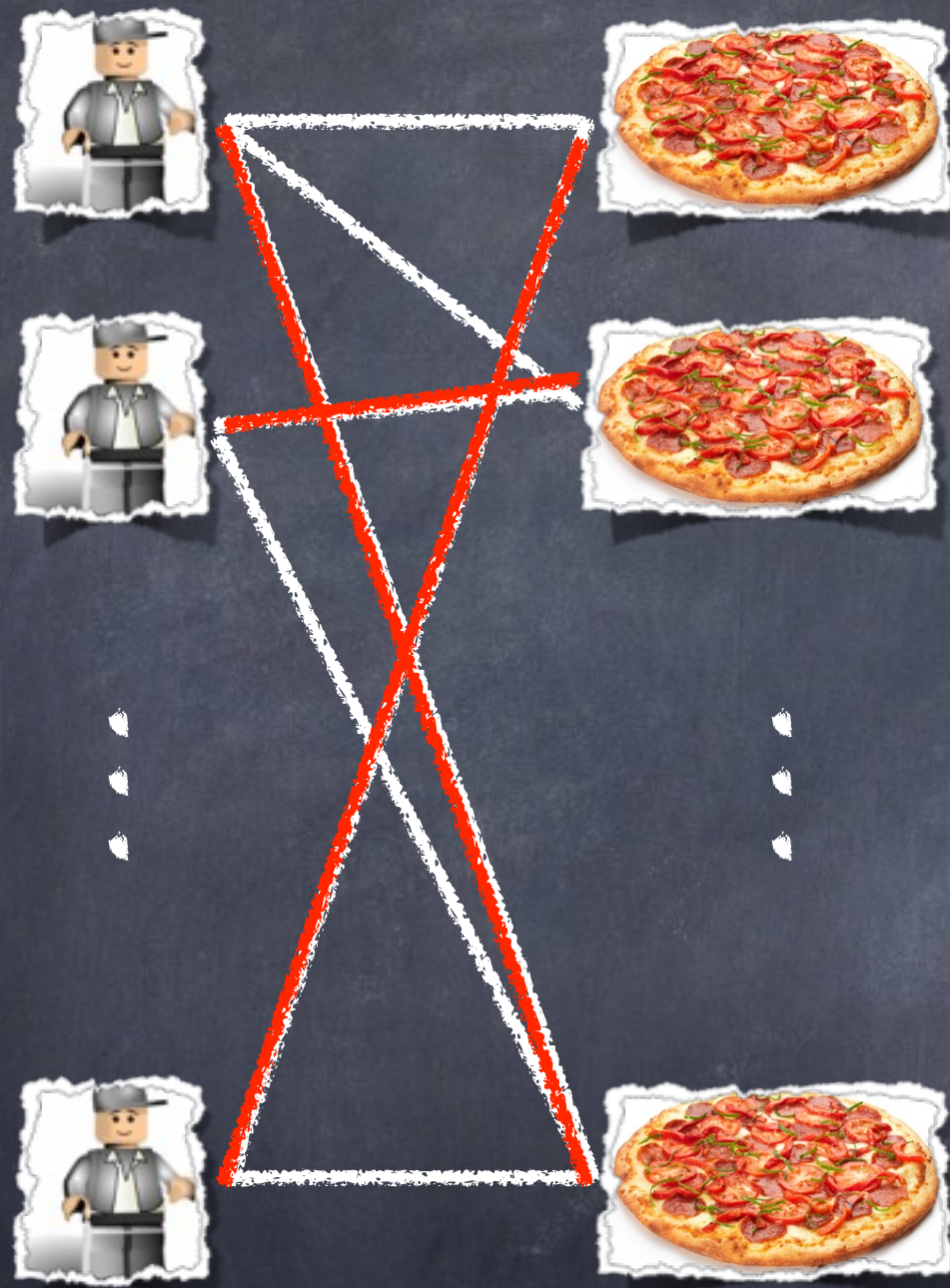
$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

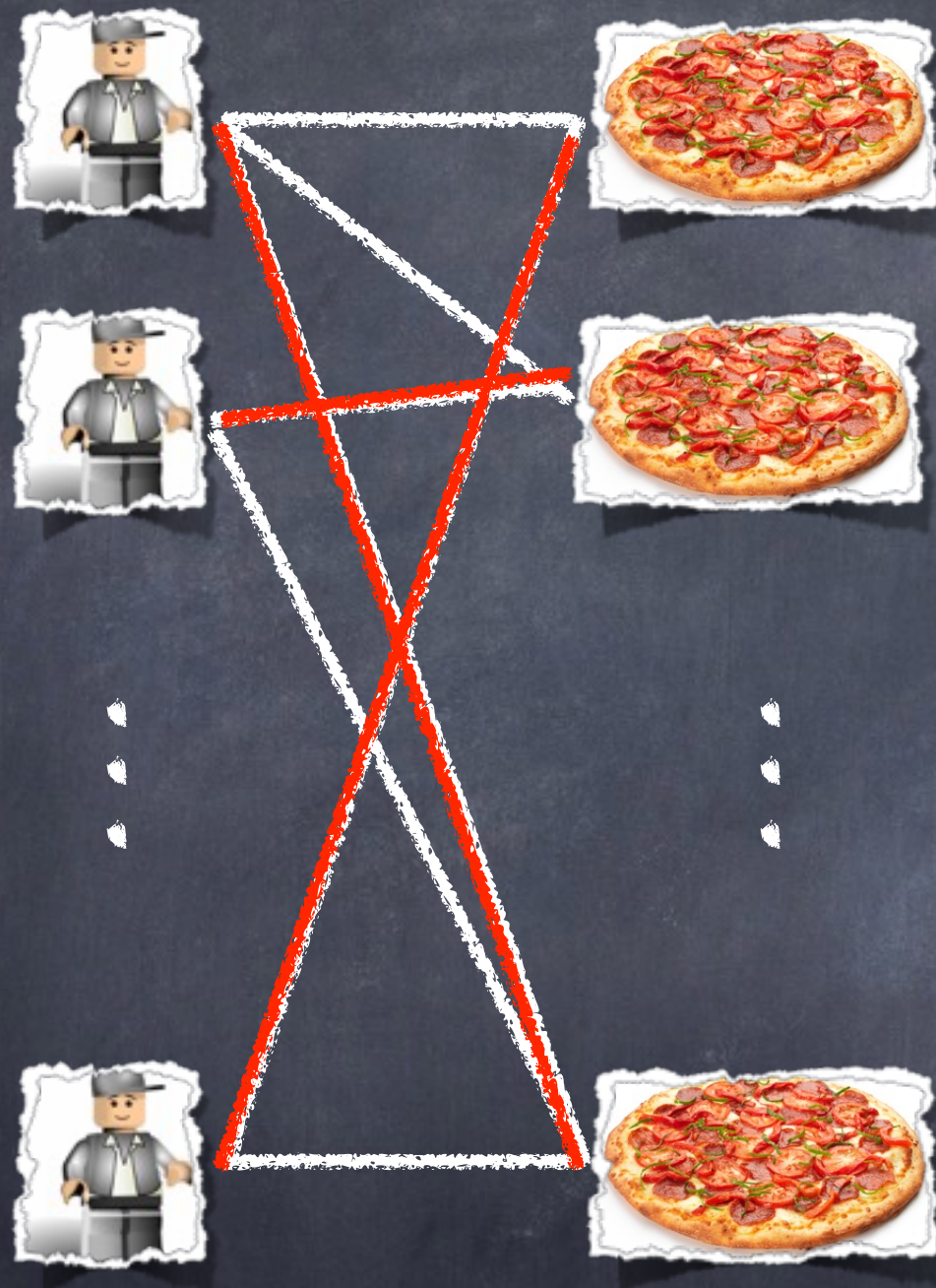
$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

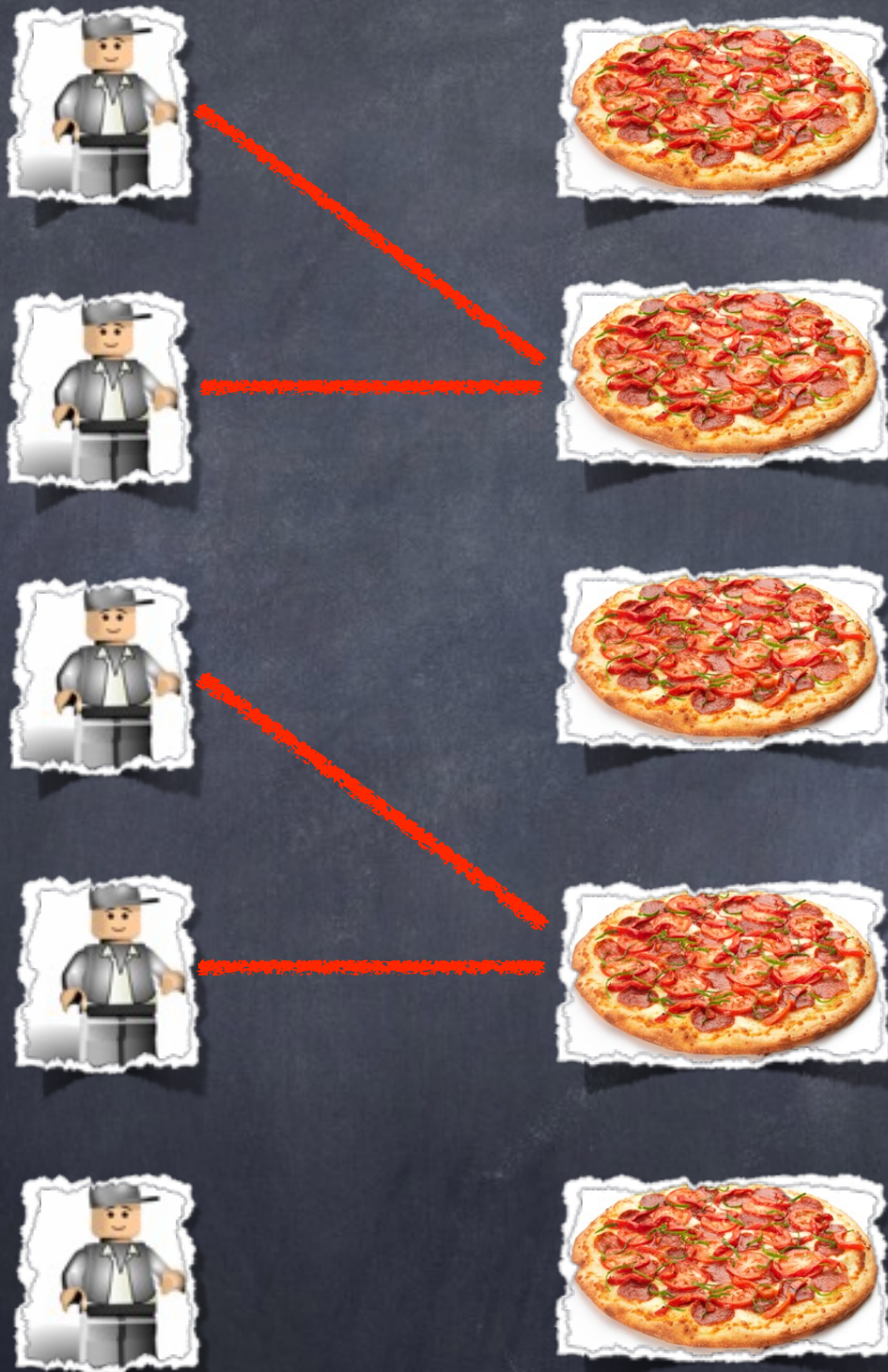
was heißt
„so lange möglich“?

wie verbessern
wir E'

Pizzabestellung

wie verbessern
wir E'

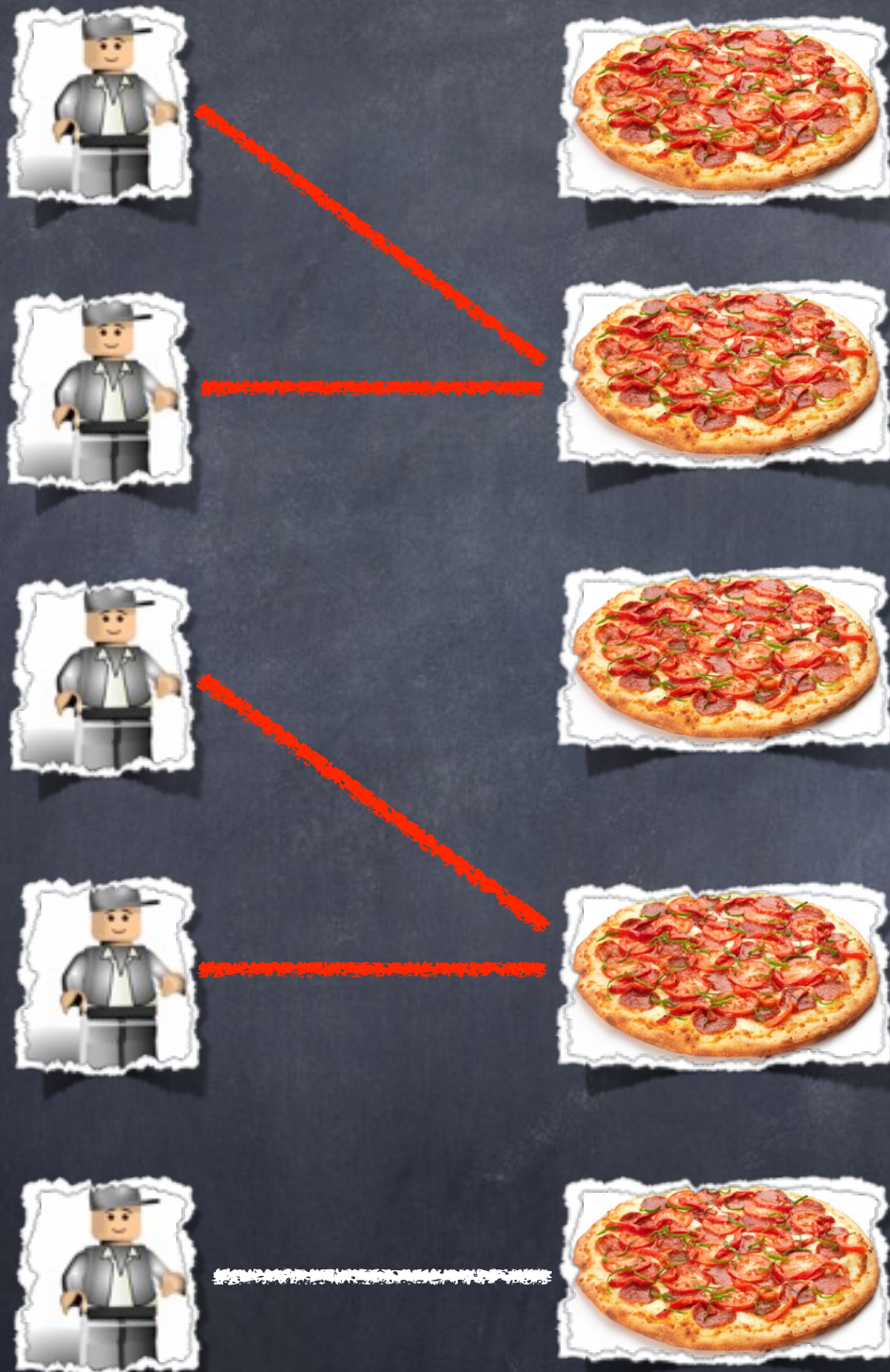
Am Bsp.:
max. Größe: 2, $n=5$, $m=5$



Pizzabestellung

wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

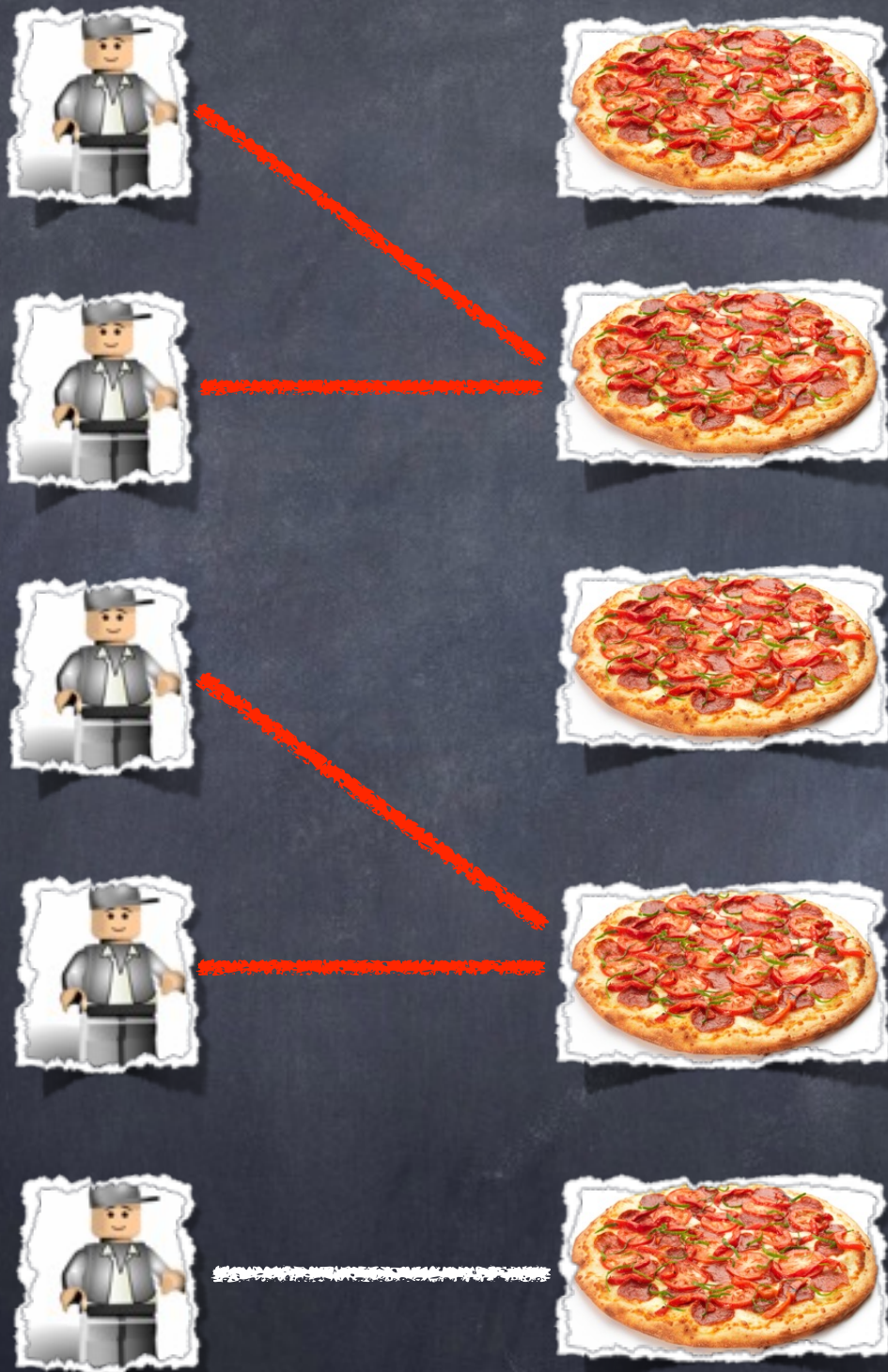


Pizzabestellung

wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

$$|E'| = 5 = n$$



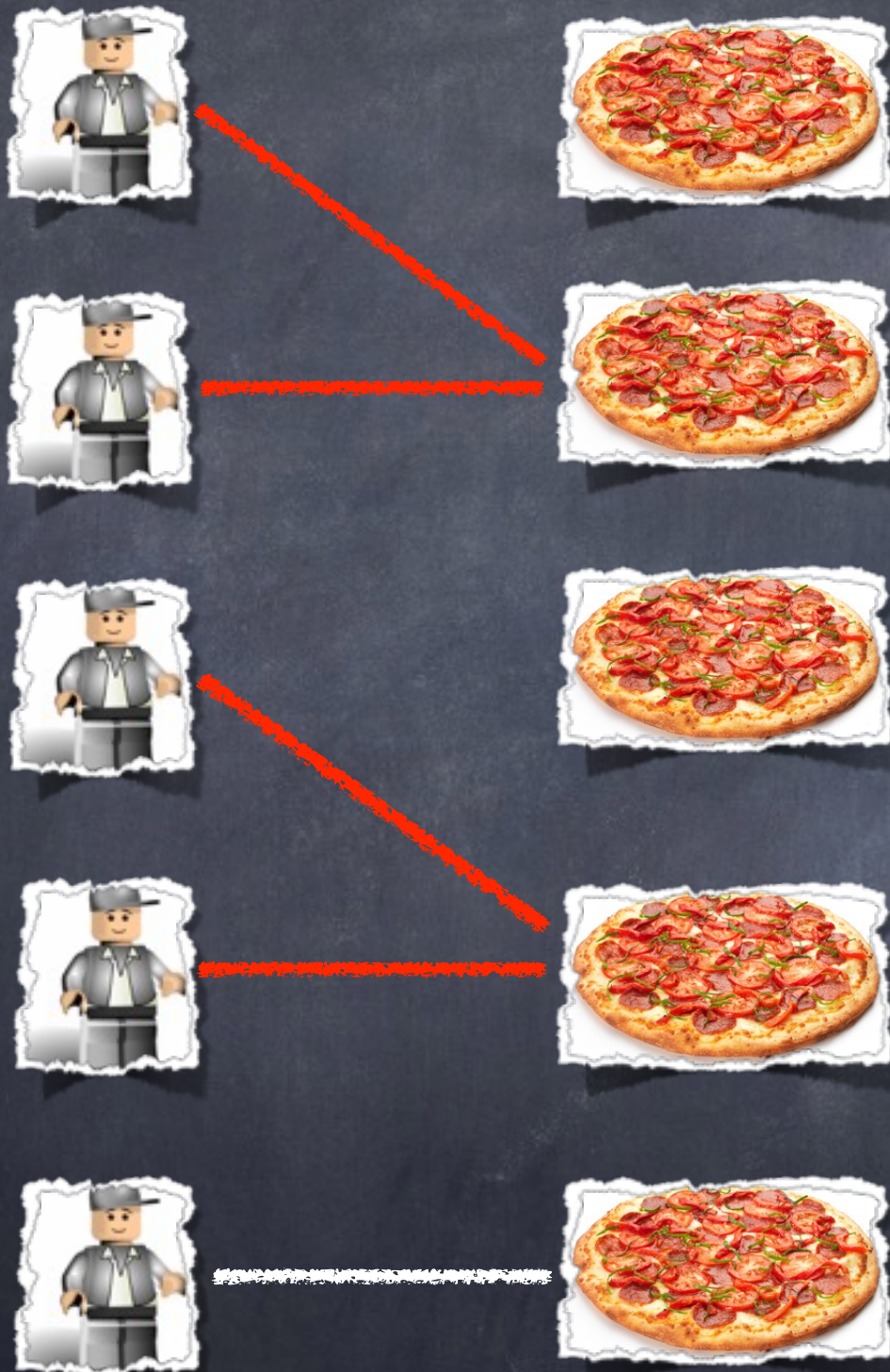
Pizzabestellung

wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

$$|E'| = 5 = n$$

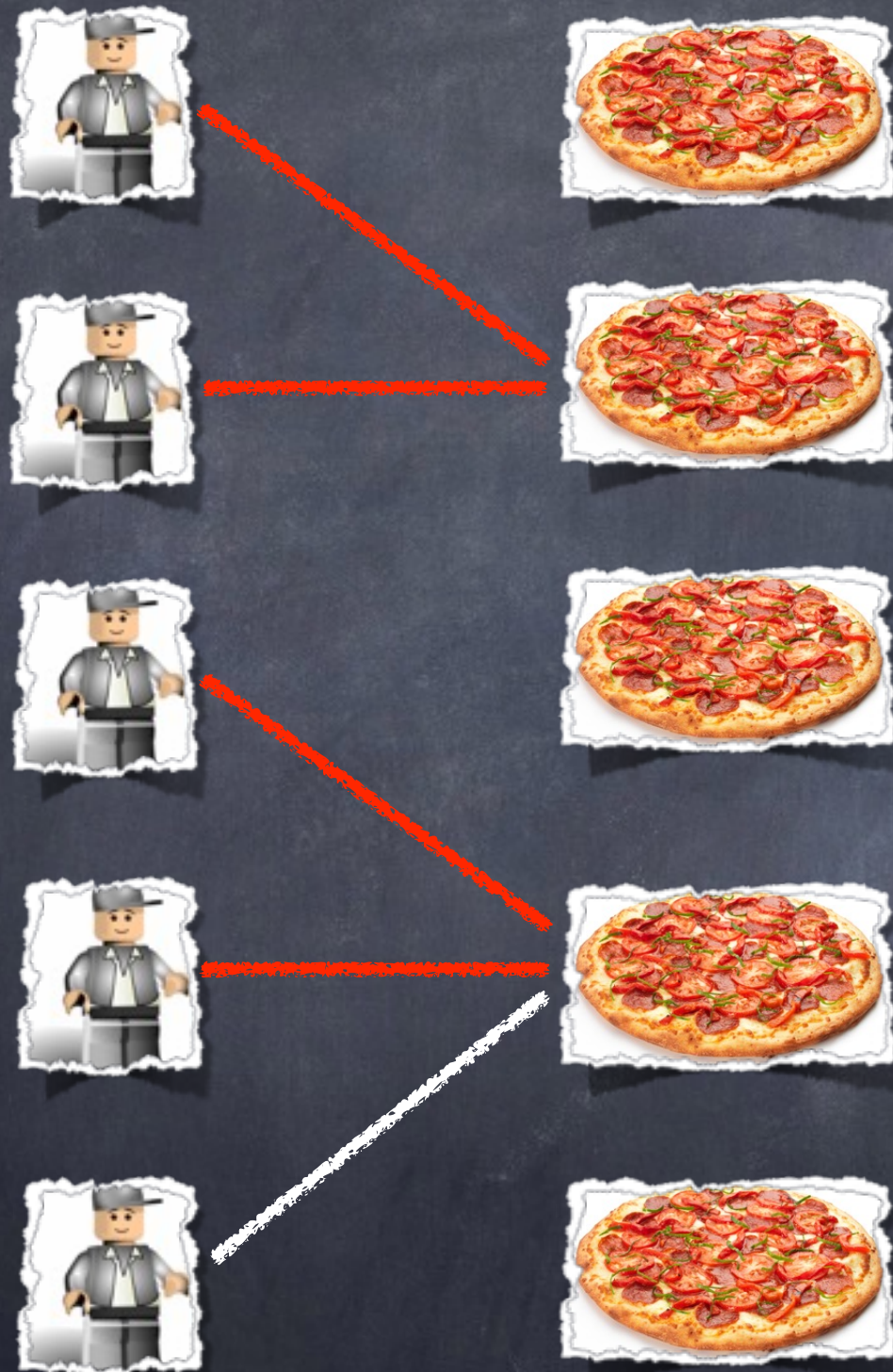
→ optimale
Einteilung



Pizzabestellung

wie verbessern
wir E'

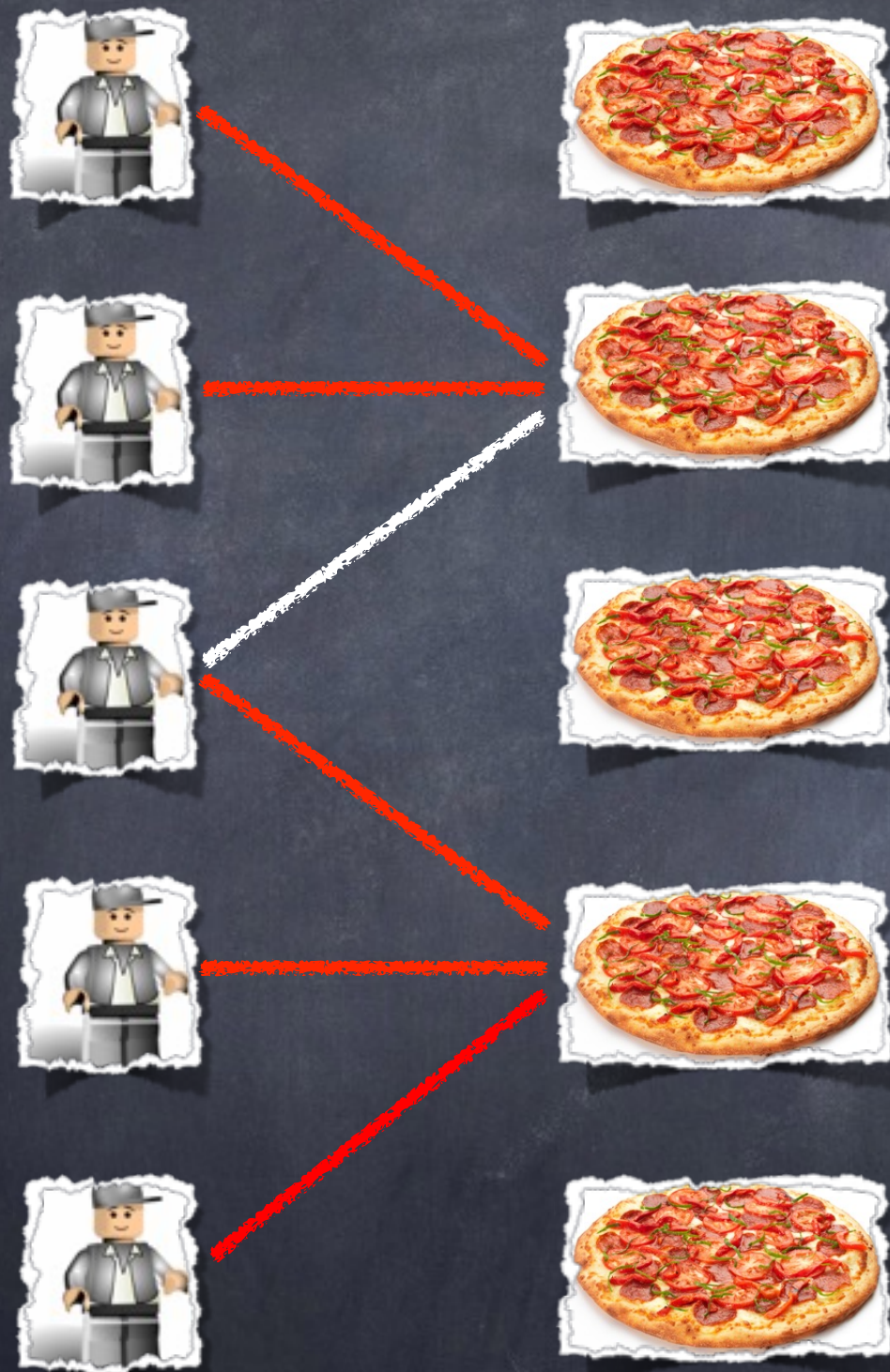
Am Bsp.:
max. Größe: 2, $n=5$, $m=5$



Pizzabestellung

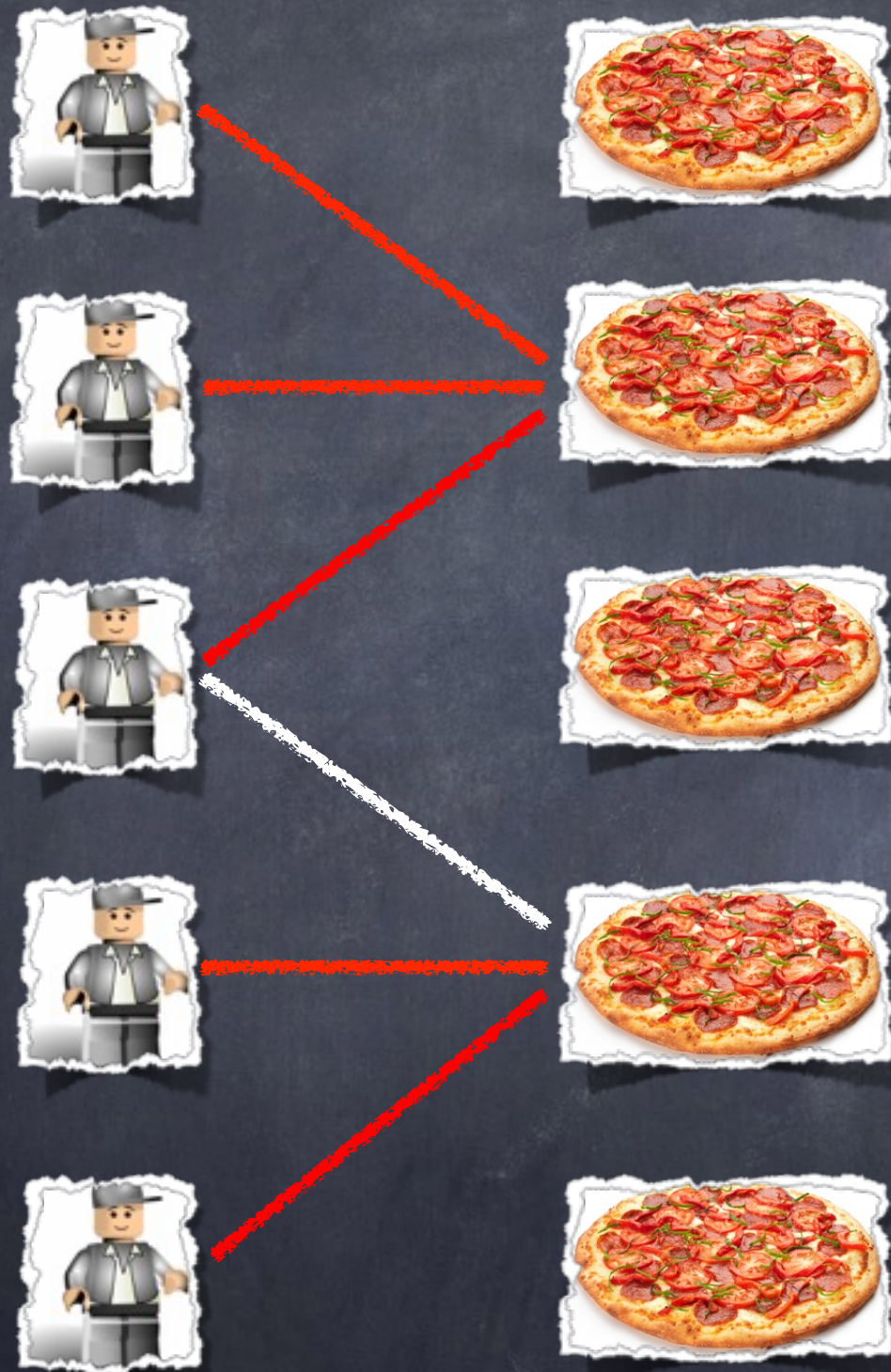
wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$



Pizzabestellung

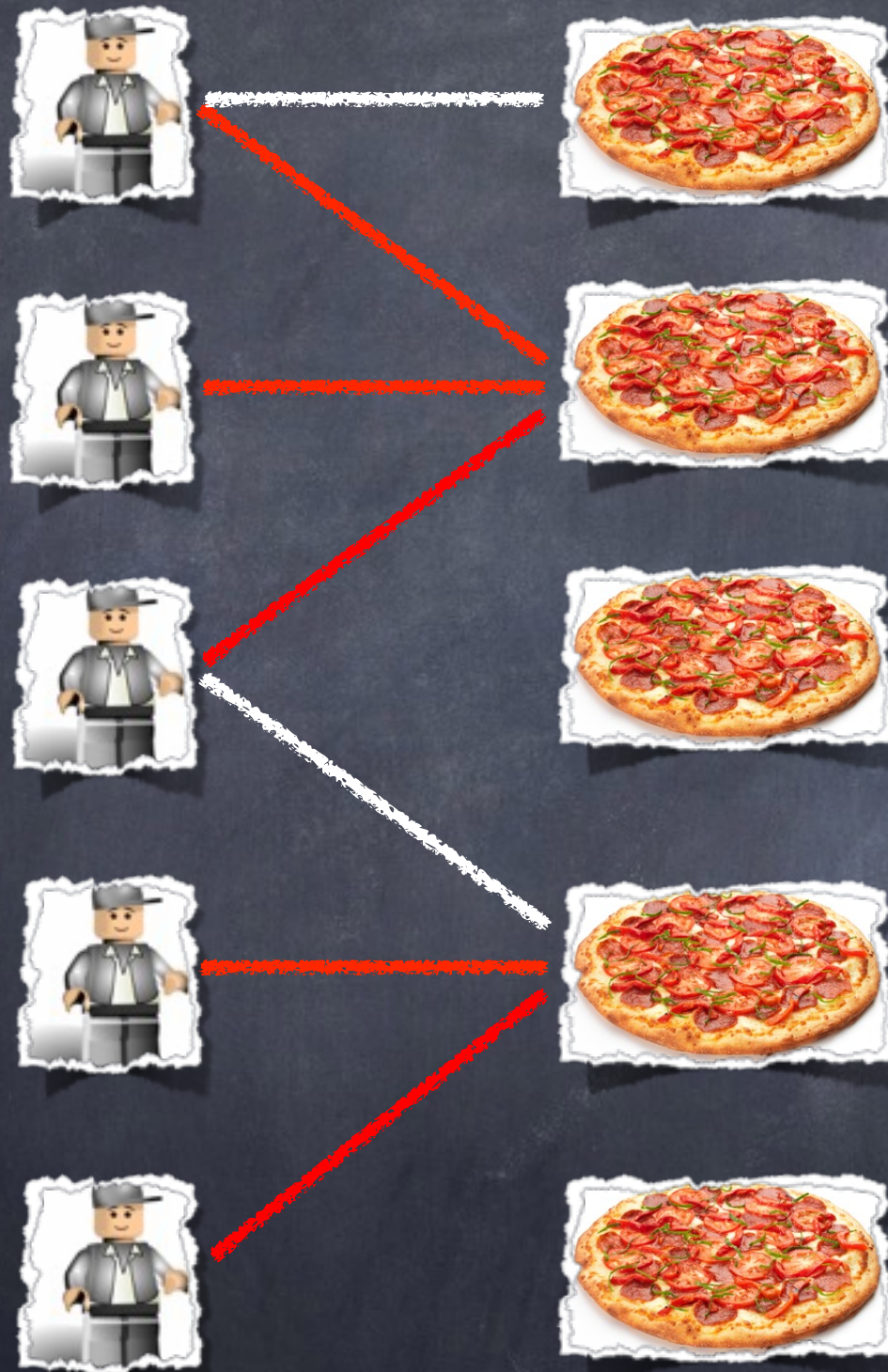
wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

Pizzabestellung

wie verbessern
wir E'

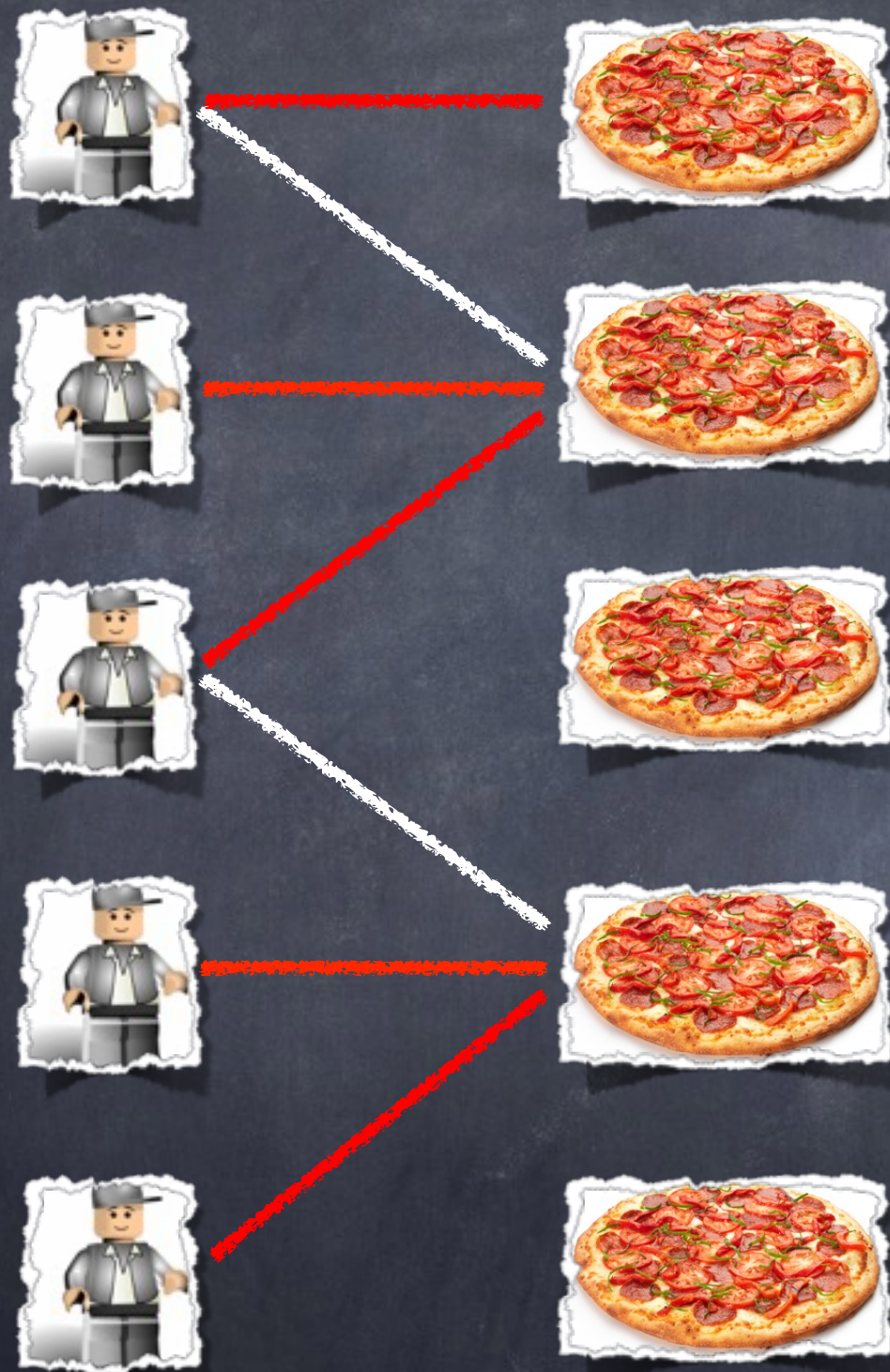


Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

Pizzabestellung

wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$



Pizzabestellung

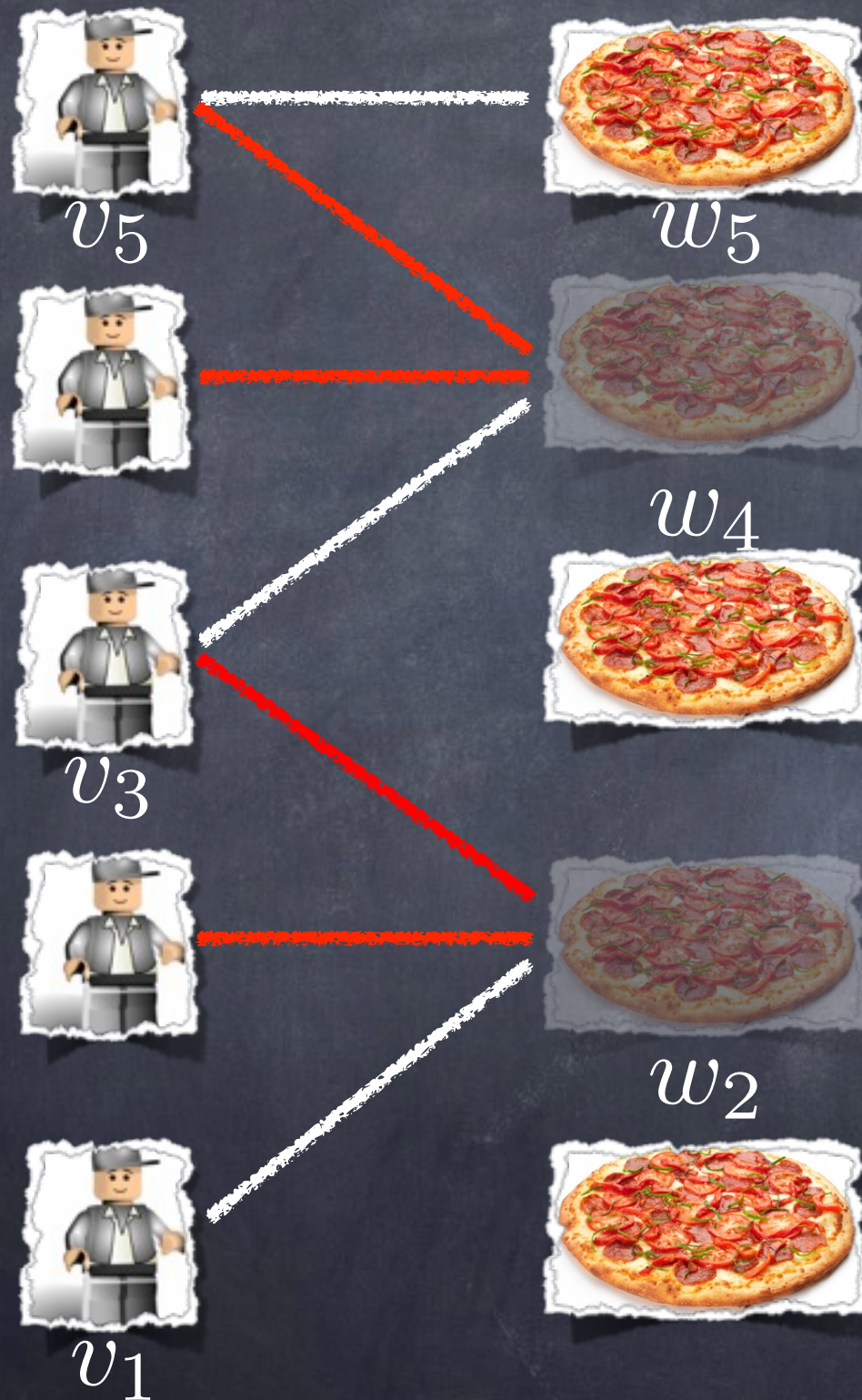
wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

Student v_3 gibt Stück
v. Pizza w_2 zurück

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

Student v_3 gibt Stück
v. Pizza w_2 zurück

Student v_3 kriegt
Stück v. Pizza w_4

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

Student v_3 gibt Stück
v. Pizza w_2 zurück

Student v_3 kriegt
Stück v. Pizza w_4

Student v_5 gibt Stück
v. Pizza w_4 zurück

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

Student v_3 gibt Stück
v. Pizza w_2 zurück

Student v_3 kriegt
Stück v. Pizza w_4

Student v_5 gibt Stück
v. Pizza w_4 zurück

Student v_5 kriegt
Stück v. Pizza w_5

Pizzabestellung



wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

laufe unbenutzte
Kante entlang

Student v_3 gibt Stück
v. Pizza w_2 zurück

Student v_3 kriegt
Stück v. Pizza w_4

Student v_5 gibt Stück
v. Pizza w_4 zurück

Student v_5 kriegt
Stück v. Pizza w_5

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

Student v_3 gibt Stück
v. Pizza w_2 zurück

Student v_3 kriegt
Stück v. Pizza w_4

Student v_5 gibt Stück
v. Pizza w_4 zurück

Student v_5 kriegt
Stück v. Pizza w_5

laufe unbenutzte
Kante entlang

laufe benutzte
Kante entlang

Pizzabestellung



wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

Student v_3 gibt Stück
v. Pizza w_2 zurück

Student v_3 kriegt
Stück v. Pizza w_4

Student v_5 gibt Stück
v. Pizza w_4 zurück

Student v_5 kriegt
Stück v. Pizza w_5

laufe unbenutzte
Kante entlang

laufe benutzte
Kante entlang

laufe unbenutzte
Kante entlang

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:

max. Größe: 2, $n=5$, $m=5$

Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

Student v_3 gibt Stück
v. Pizza w_2 zurück

Student v_3 kriegt
Stück v. Pizza w_4

Student v_5 gibt Stück
v. Pizza w_4 zurück

Student v_5 kriegt
Stück v. Pizza w_5

laufe unbenutzte
Kante entlang

laufe benutzte
Kante entlang

laufe unbenutzte
Kante entlang

laufe benutzte
Kante entlang

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

laufe unbenutzte
Kante entlang

Student v_3 gibt Stück
v. Pizza w_2 zurück

laufe benutzte
Kante entlang

Student v_3 kriegt
Stück v. Pizza w_4

laufe unbenutzte
Kante entlang

Student v_5 gibt Stück
v. Pizza w_4 zurück

laufe benutzte
Kante entlang

Student v_5 kriegt
Stück v. Pizza w_5

laufe unbenutzte
Kante entlang

Pizzabestellung

wie verbessern
wir E'



Am Bsp.:
max. Größe: 2, $n=5$, $m=5$
Sequenz:

Student v_1 kriegt
Stück v. Pizza w_2

laufe unbenutzte
Kante entlang

Student v_3 gibt Stück
v. Pizza w_2 zurück

laufe benutzte
Kante entlang

Student v_3 kriegt
Stück v. Pizza w_4

laufe unbenutzte
Kante entlang

Student v_5 gibt Stück
v. Pizza w_4 zurück

laufe benutzte
Kante entlang

Student v_5 kriegt
Stück v. Pizza w_5

laufe unbenutzte
Kante entlang

Pizzabestellung

wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$



Suche nach verbesserndem alt. Pfad

Pizzabestellung

wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

Student kriegt Pizzastück



Suche nach verbesserndem alt. Pfad

Pizzabestellung



wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

Student kriegt Pizzastück

Student gibt Pizzastück
zurück

Suche nach verbesserndem alt. Pfad

Pizzabestellung



Auslastung aller Pizzen (bis auf letzte)
bleibt gleich, da alternierend

wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

Student kriegt Pizzastück

Student gibt Pizzastück
zurück

Suche nach verbesserndem alt. Pfad

Pizzabestellung



wie verbessern
wir E'

Am Bsp.:

max. Größe: 2, $n=5$, $m=5$

Laufe unbenutzte Kanten e
d.h. $(v, w) = e \notin E'$
von Student v
zu Pizza w

Laufe benutzte Kanten e
d.h. $(v, w) = e \in E'$
von Pizza w
zu Student v

Pizzabestellung



wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

Laufe unbenutzte Kanten e
d.h. $(v, w) = e \notin E'$
von Student v
zu Pizza w

Laufe benutzte Kanten e
d.h. $(v, w) = e \in E'$
von Pizza w
zu Student v

Auslastung aller Pizzen (bis auf letzte)
bleibt gleich, da alternierend

Pizzabestellung



Orientiere alle Kanten und suche
verbessernden Pfad in ger. Graph

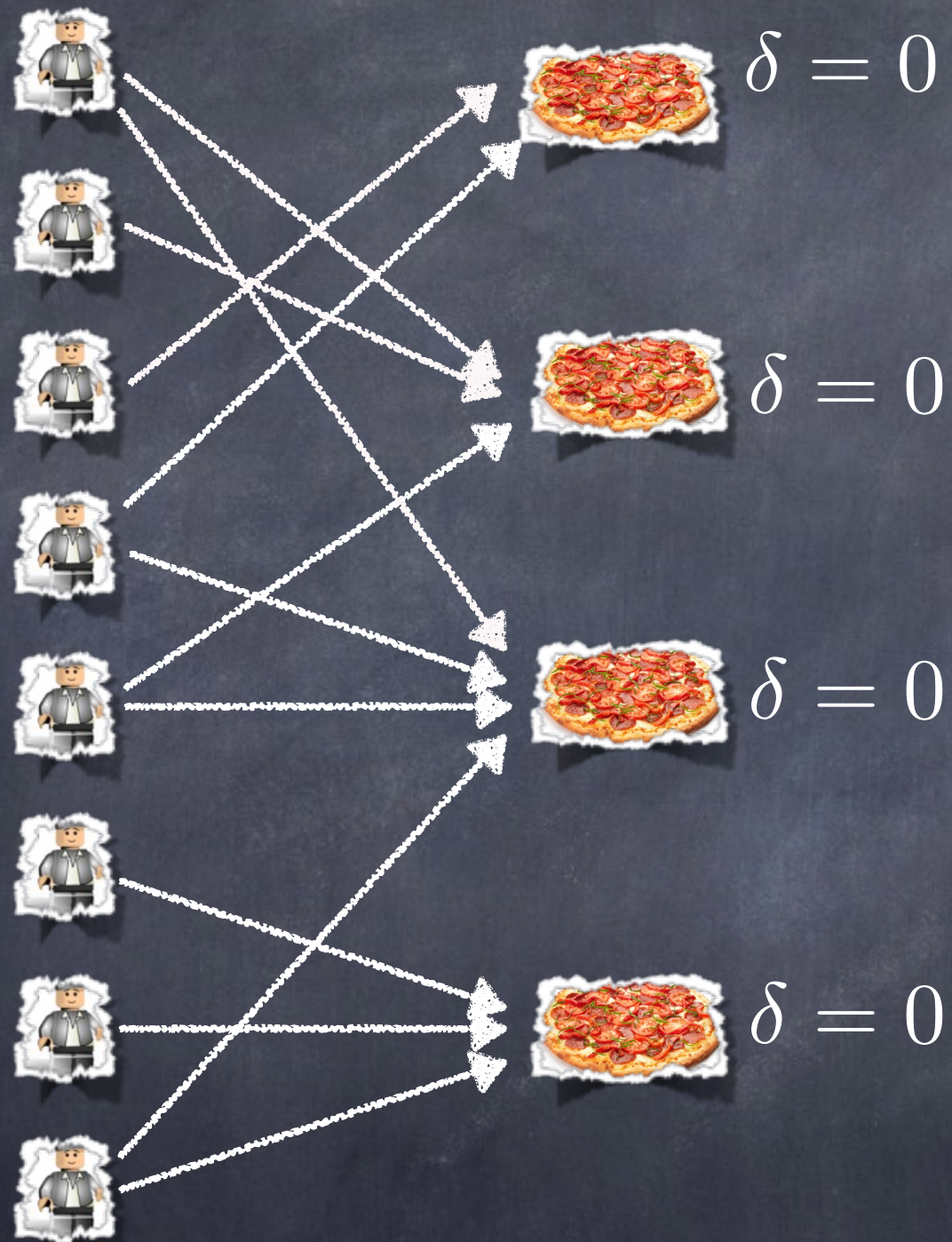
wie verbessern
wir E'

Am Bsp.:
max. Größe: 2, $n=5$, $m=5$

Laufe unbenutzte Kanten e
d.h. $(v, w) = e \notin E'$
von Student v
zu Pizza w

Laufe benutzte Kanten e
d.h. $(v, w) = e \in E'$
von Pizza w
zu Student v

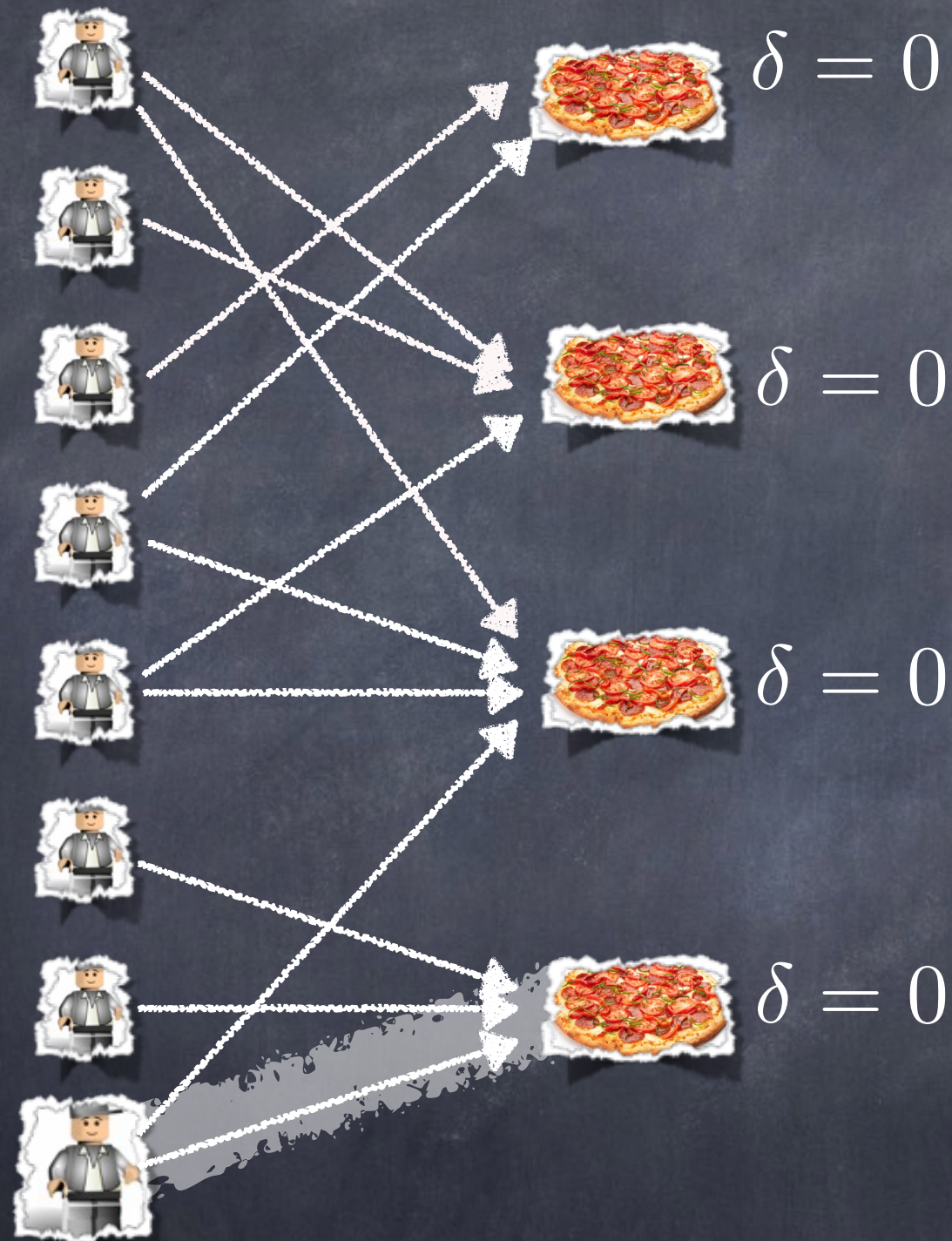
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

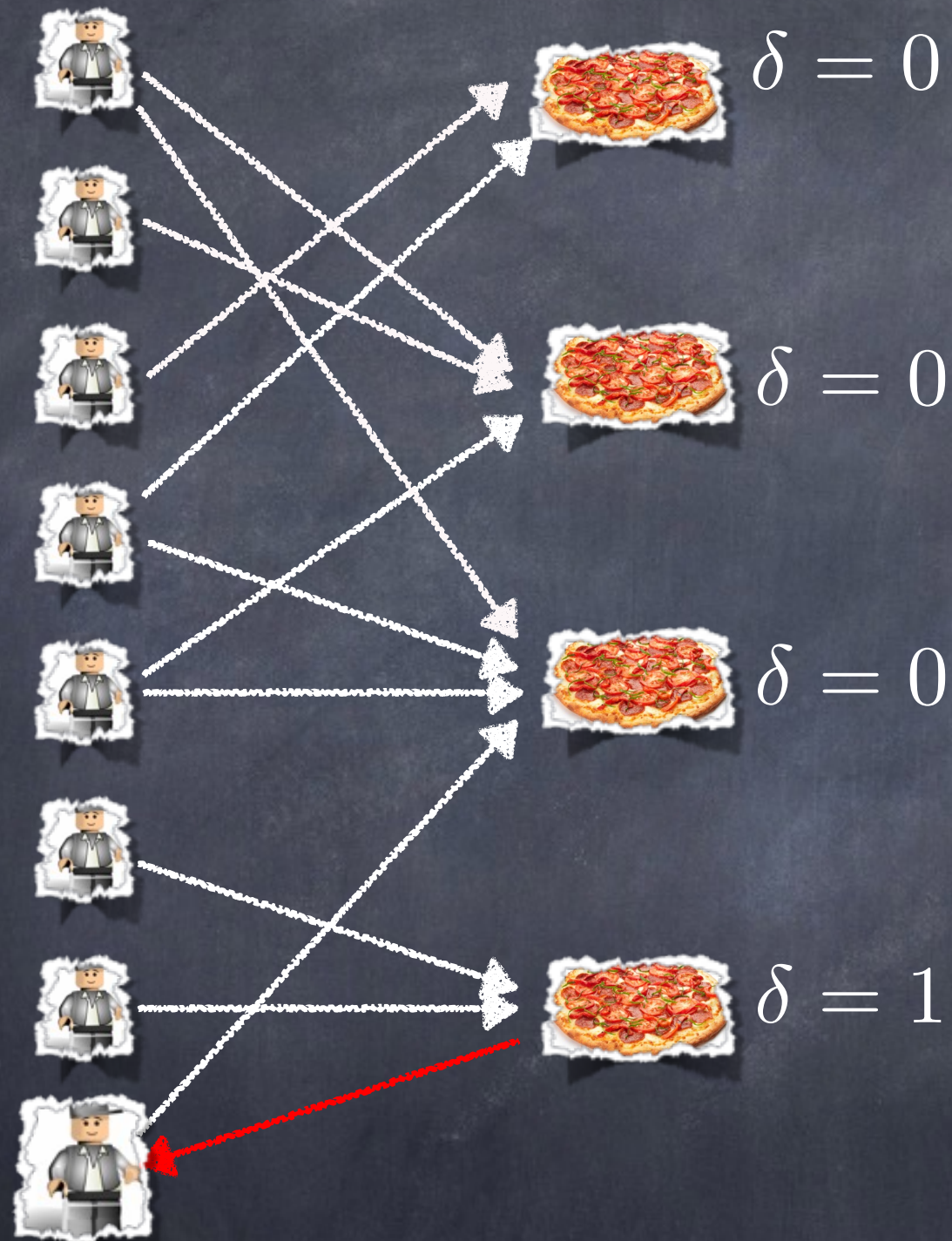
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

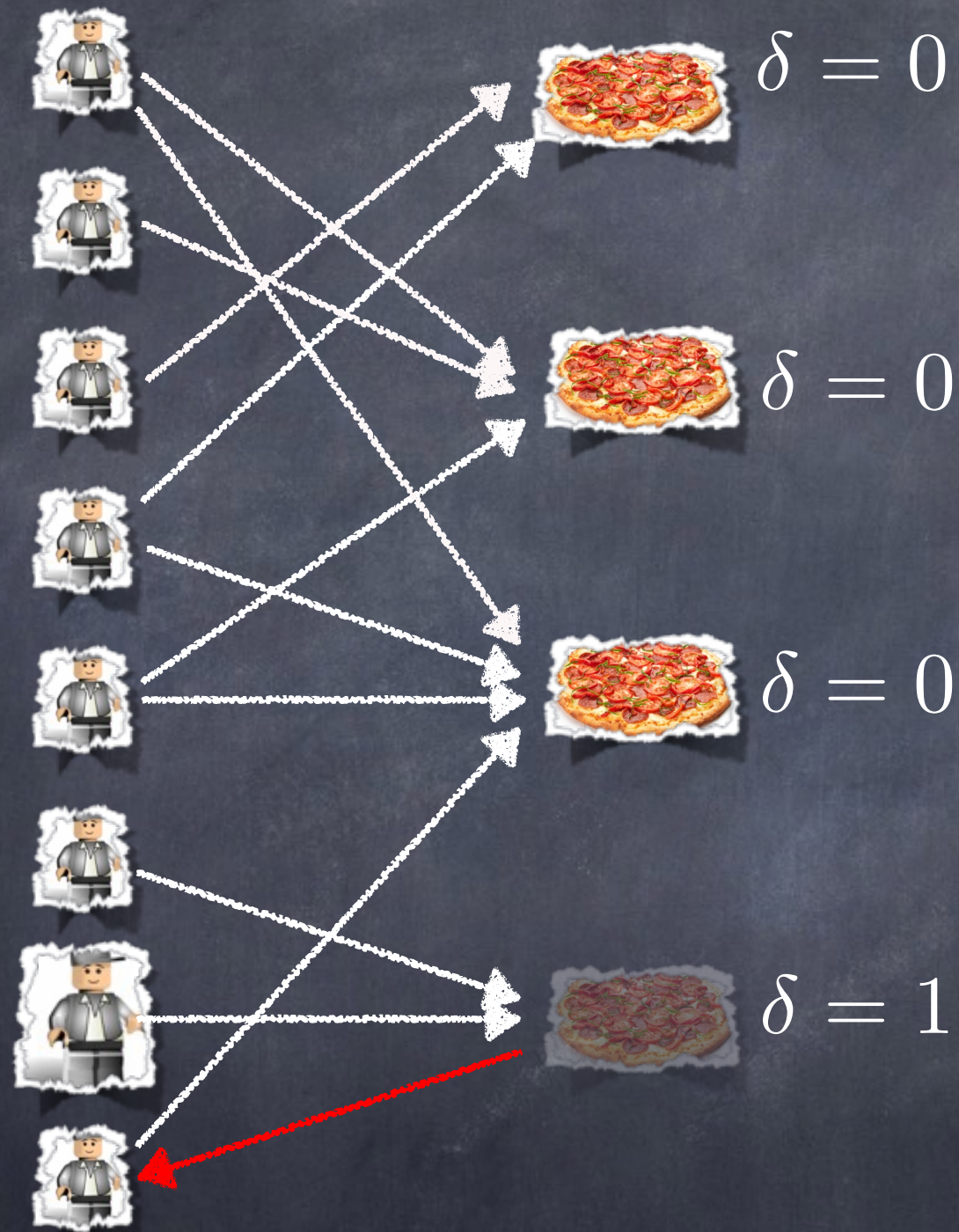
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

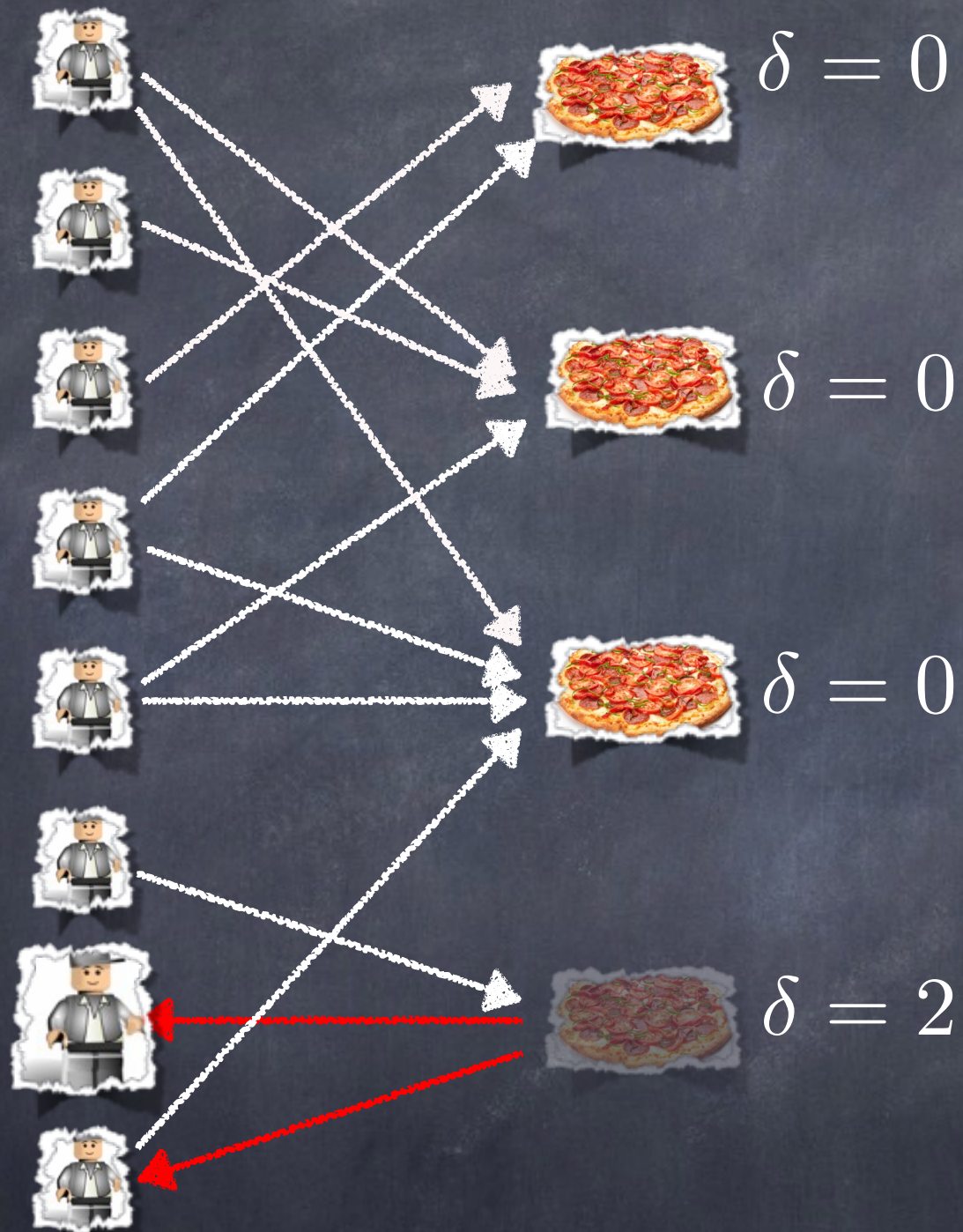
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

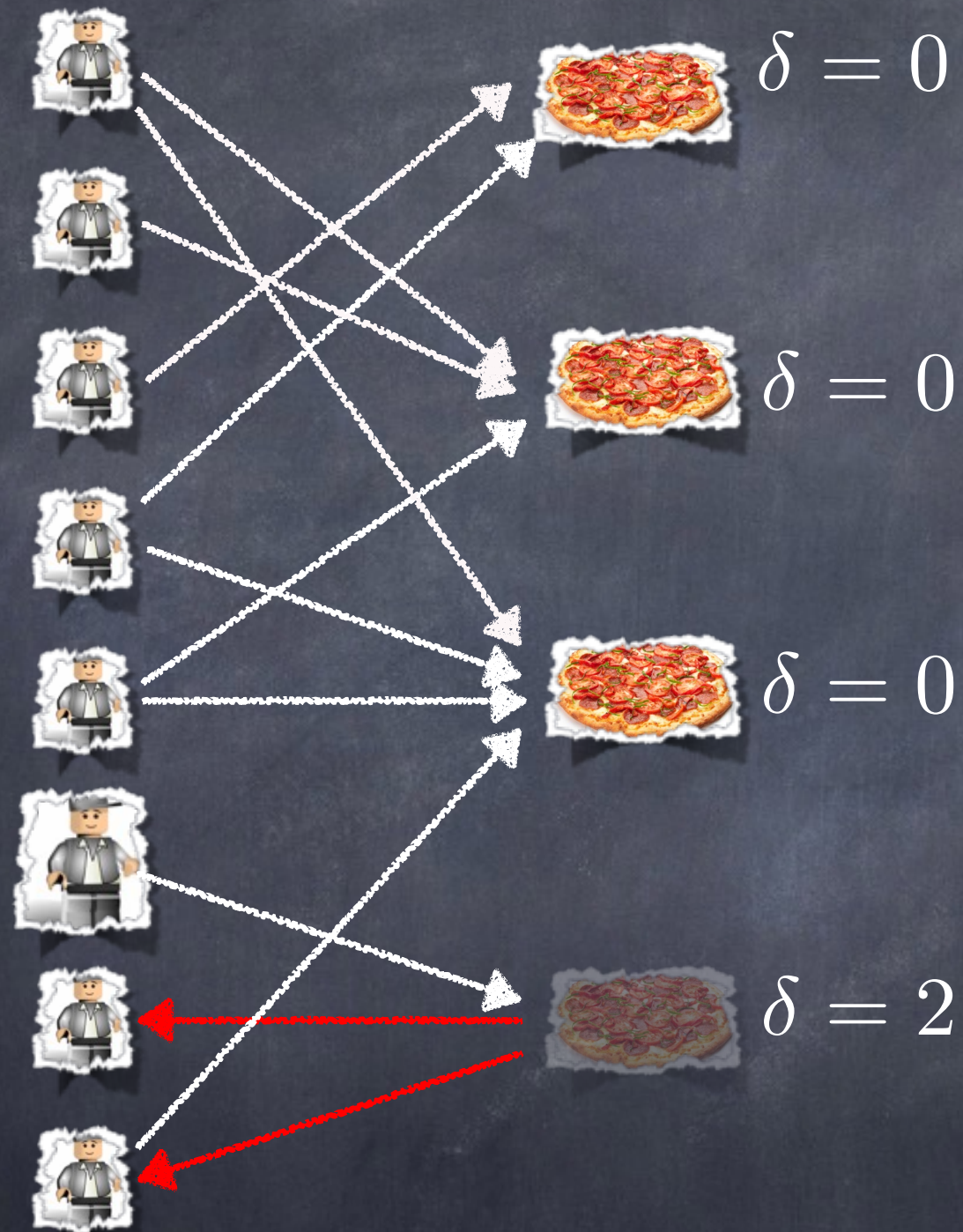
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

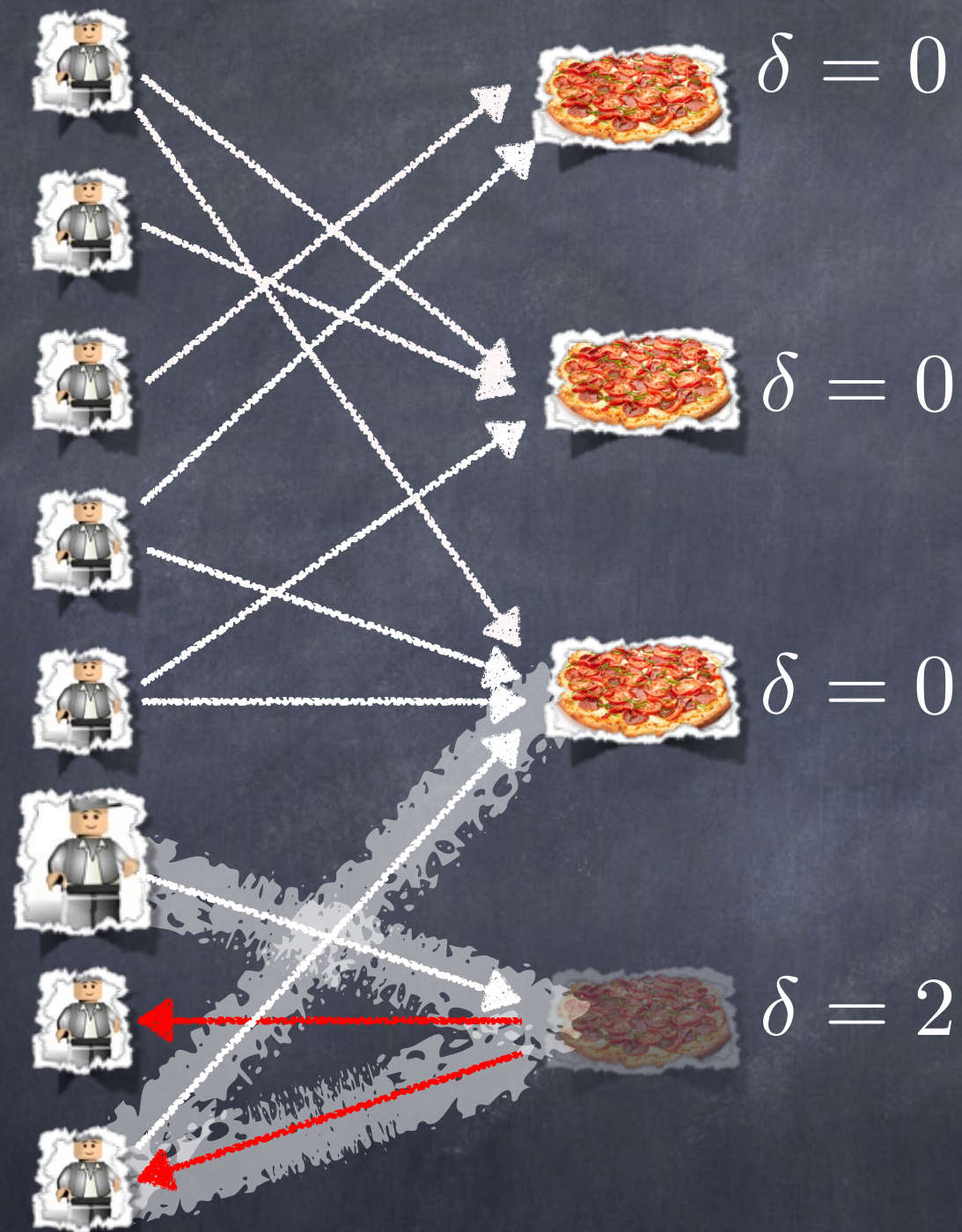
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

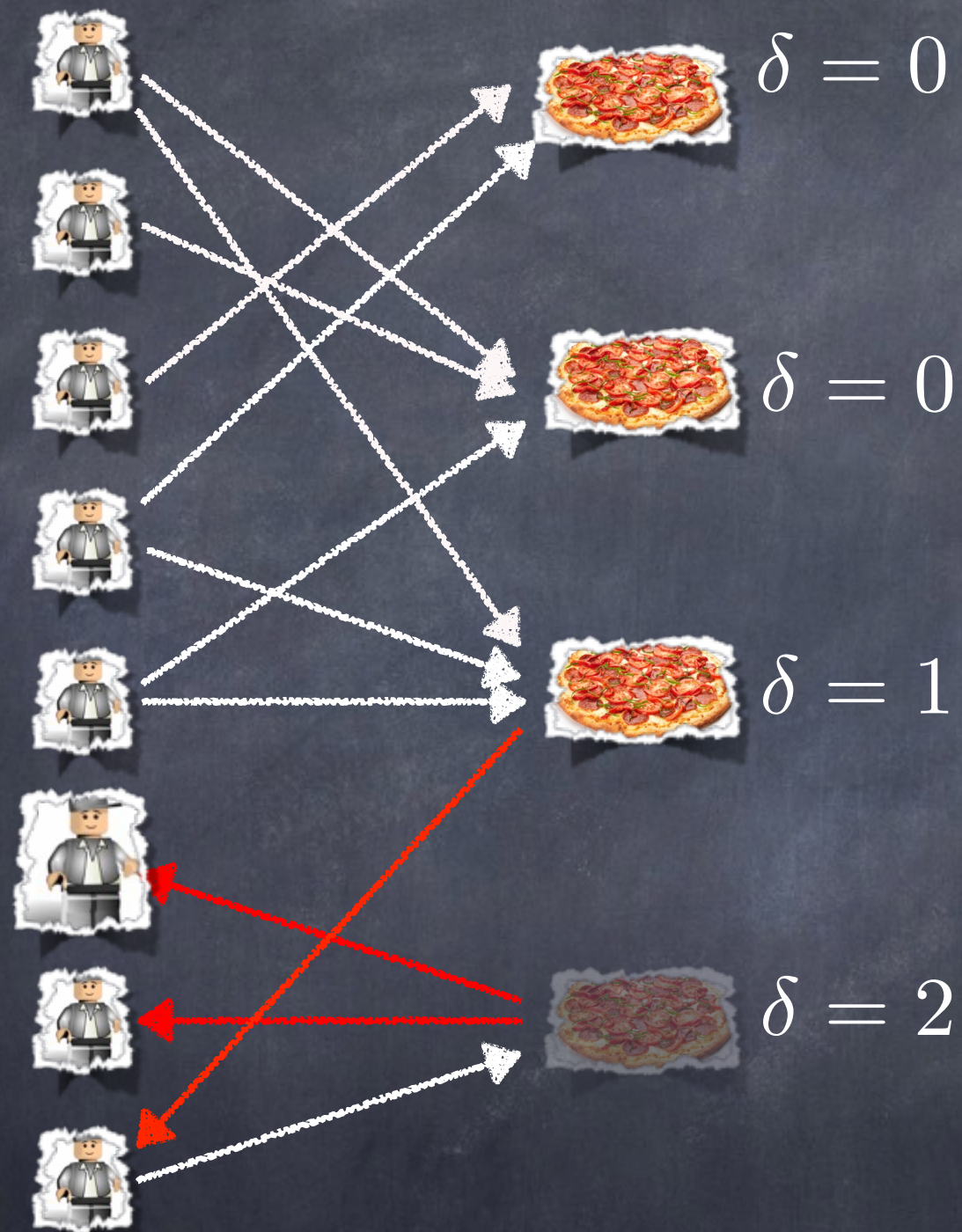
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

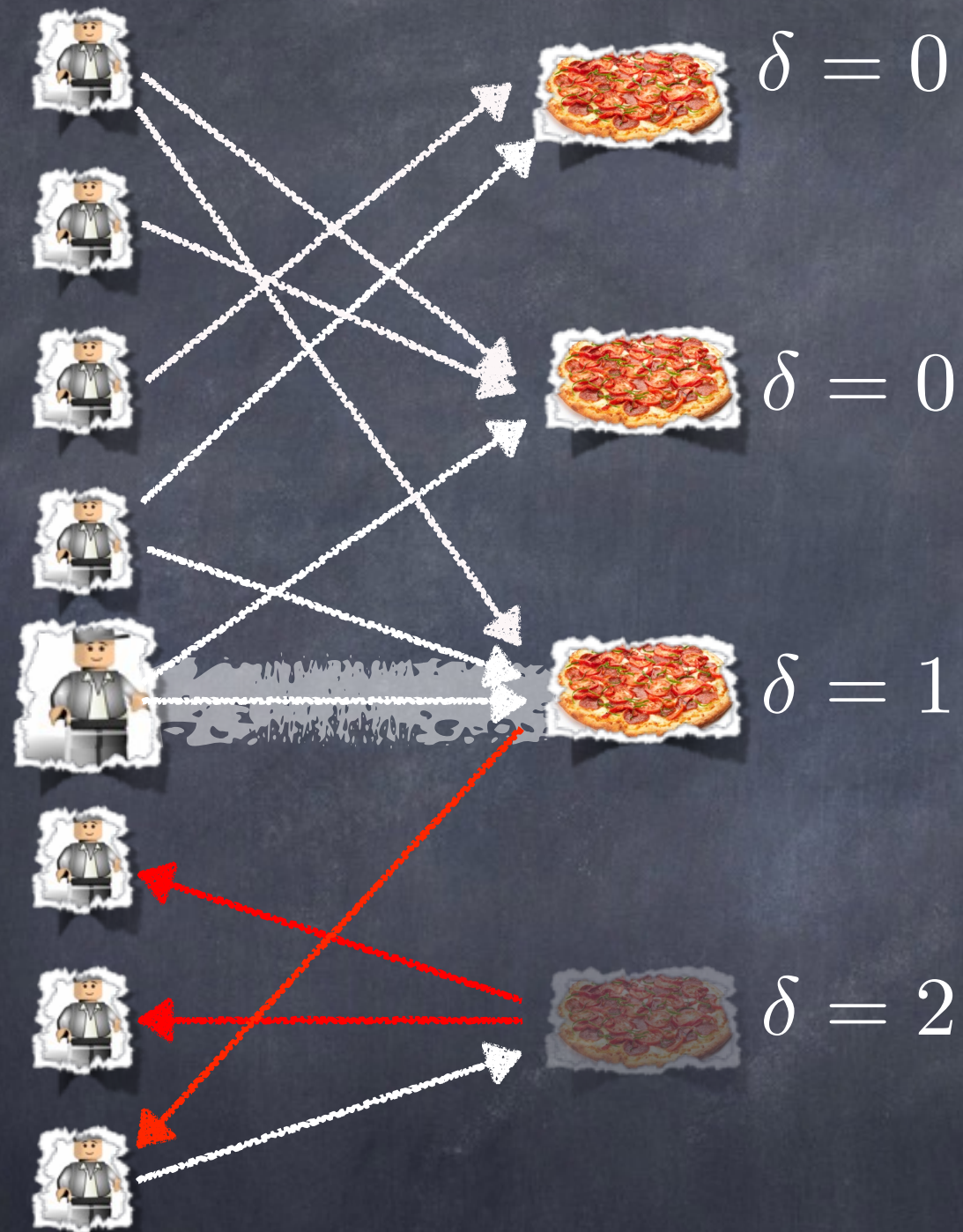
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

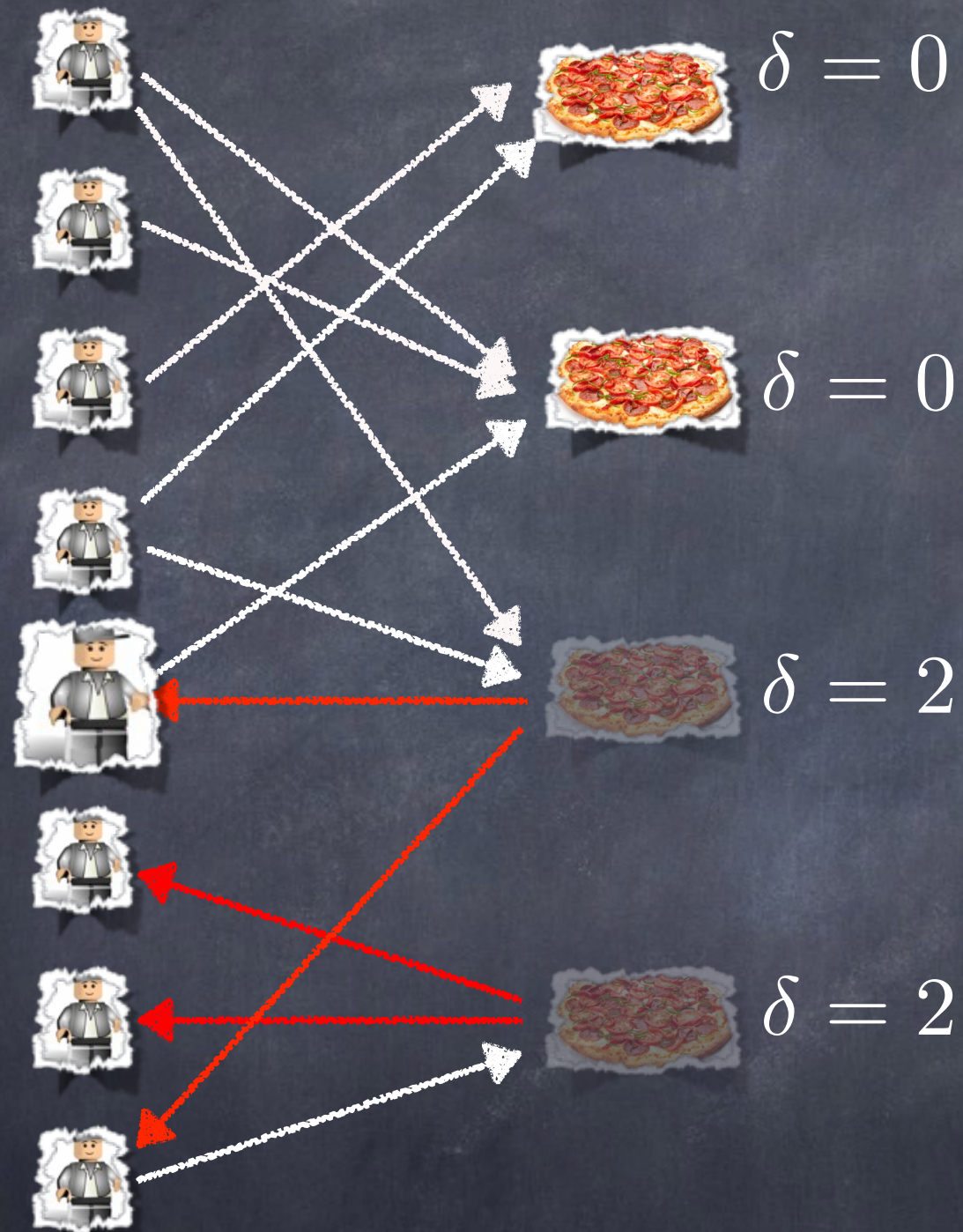
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

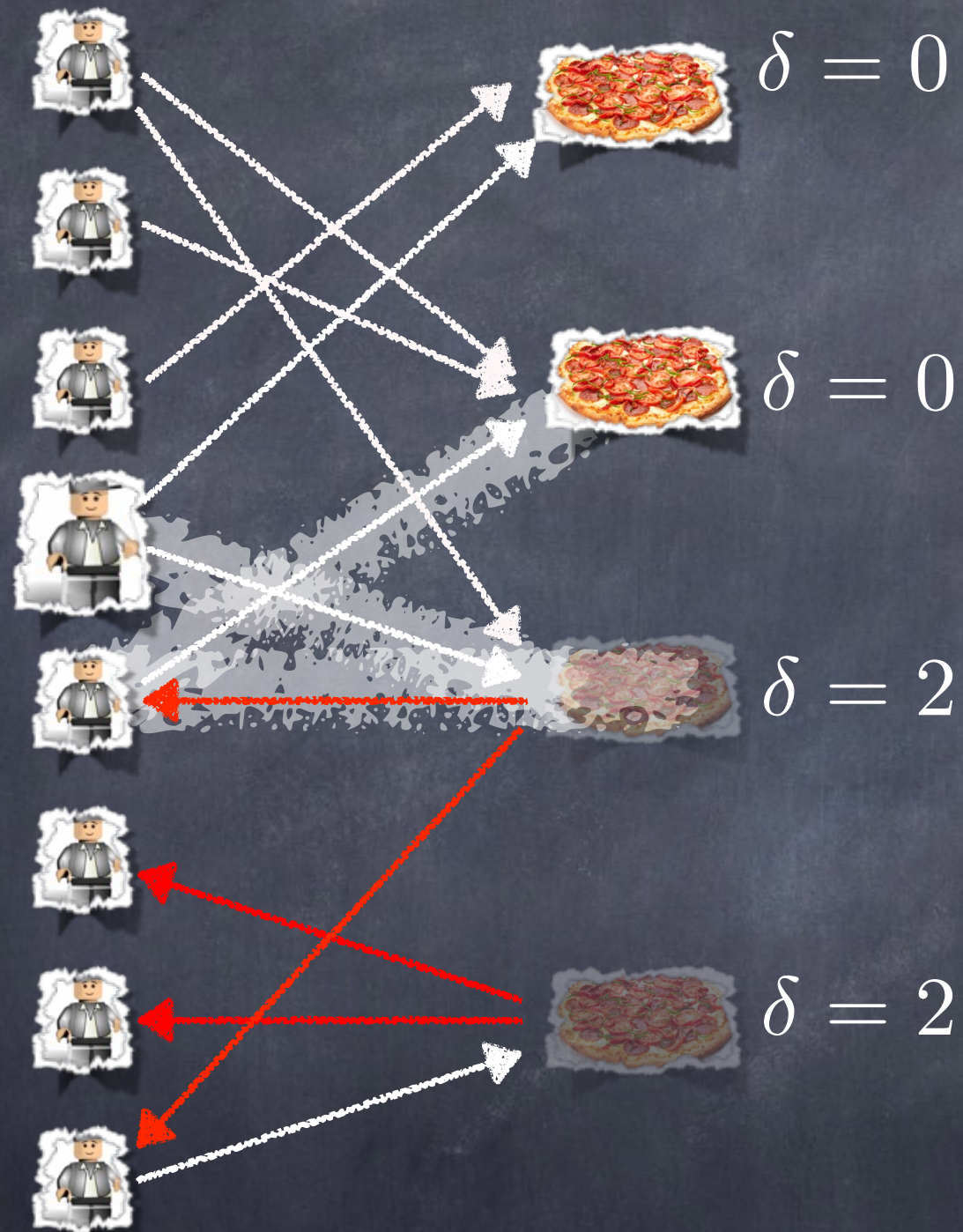
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

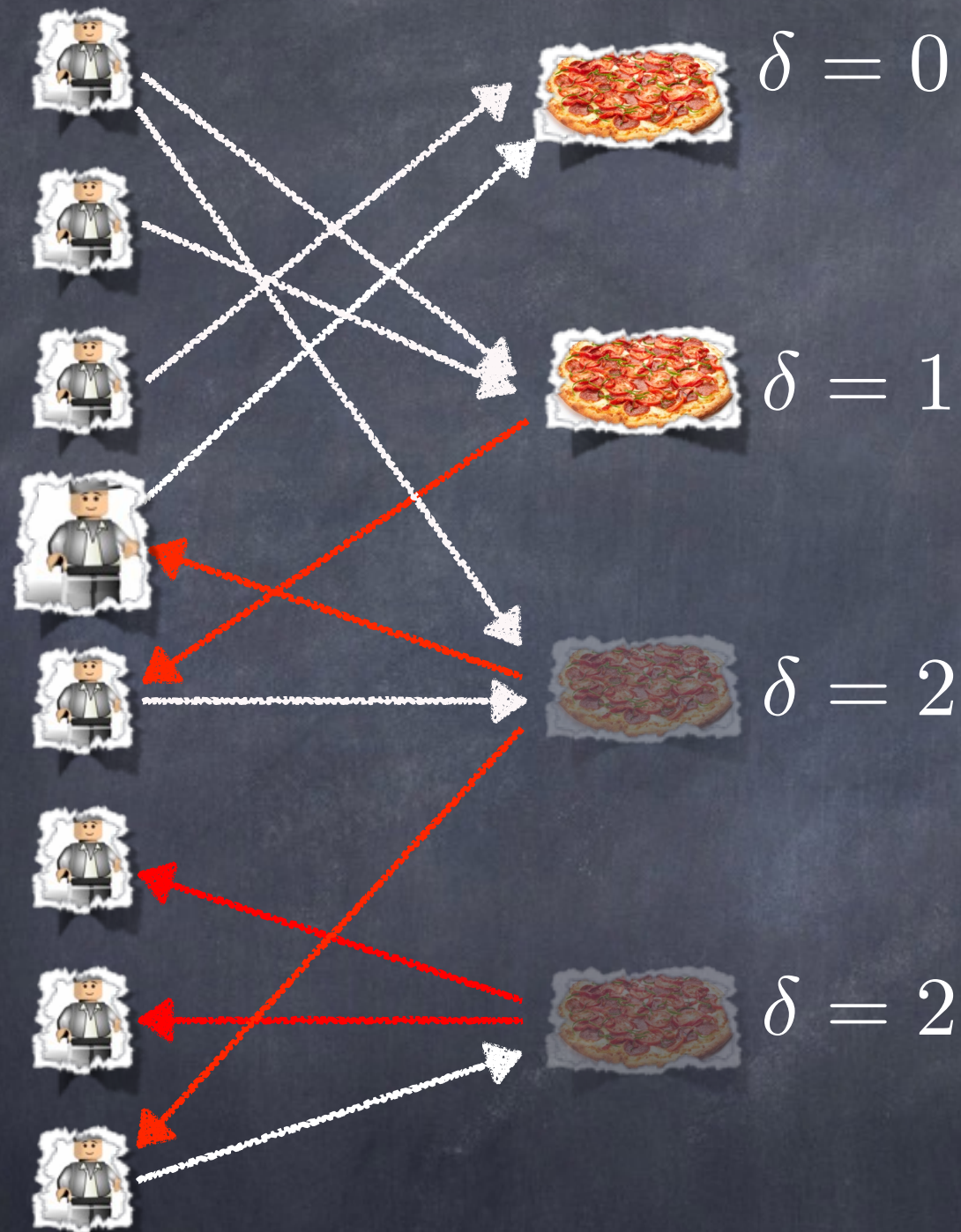
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

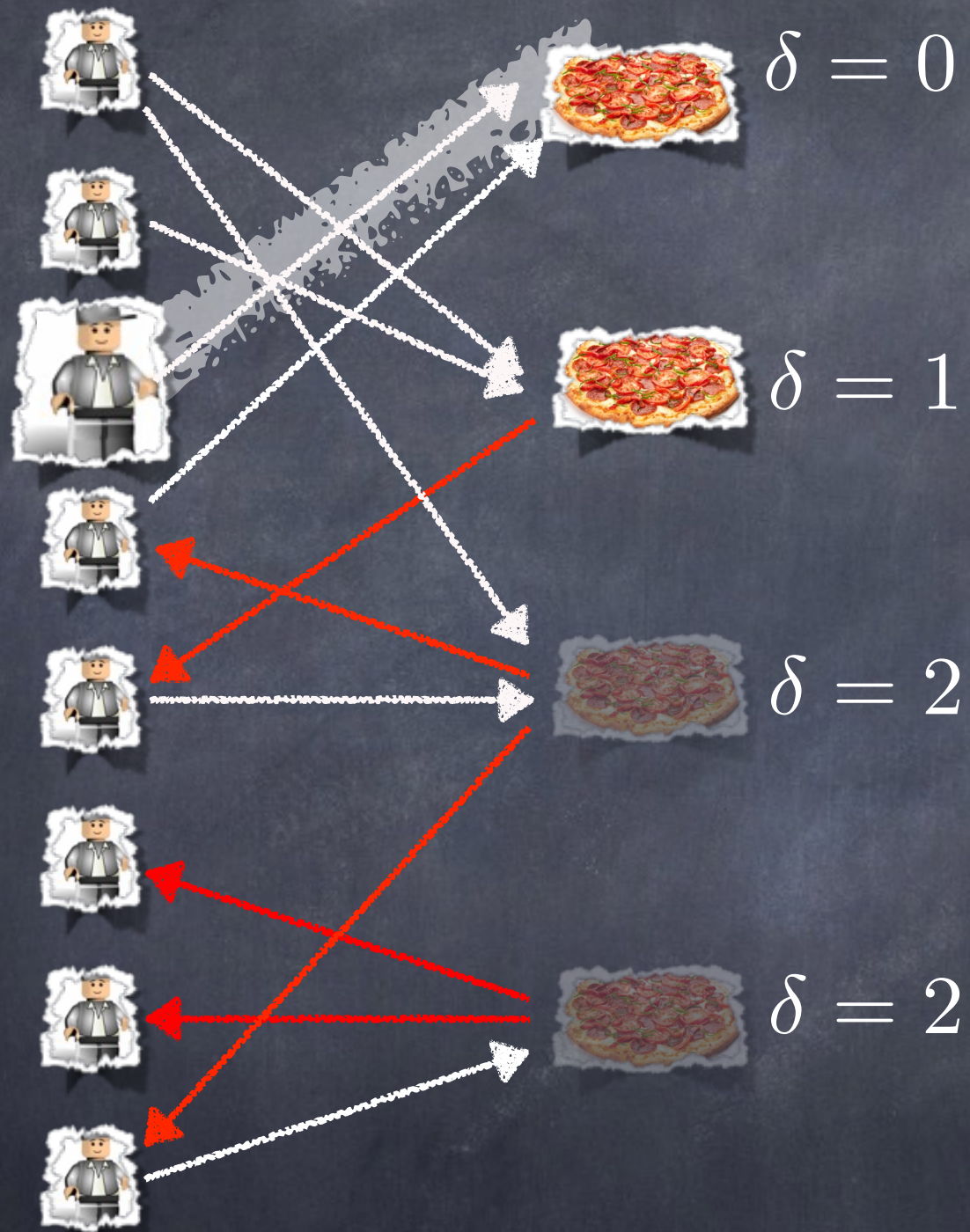
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

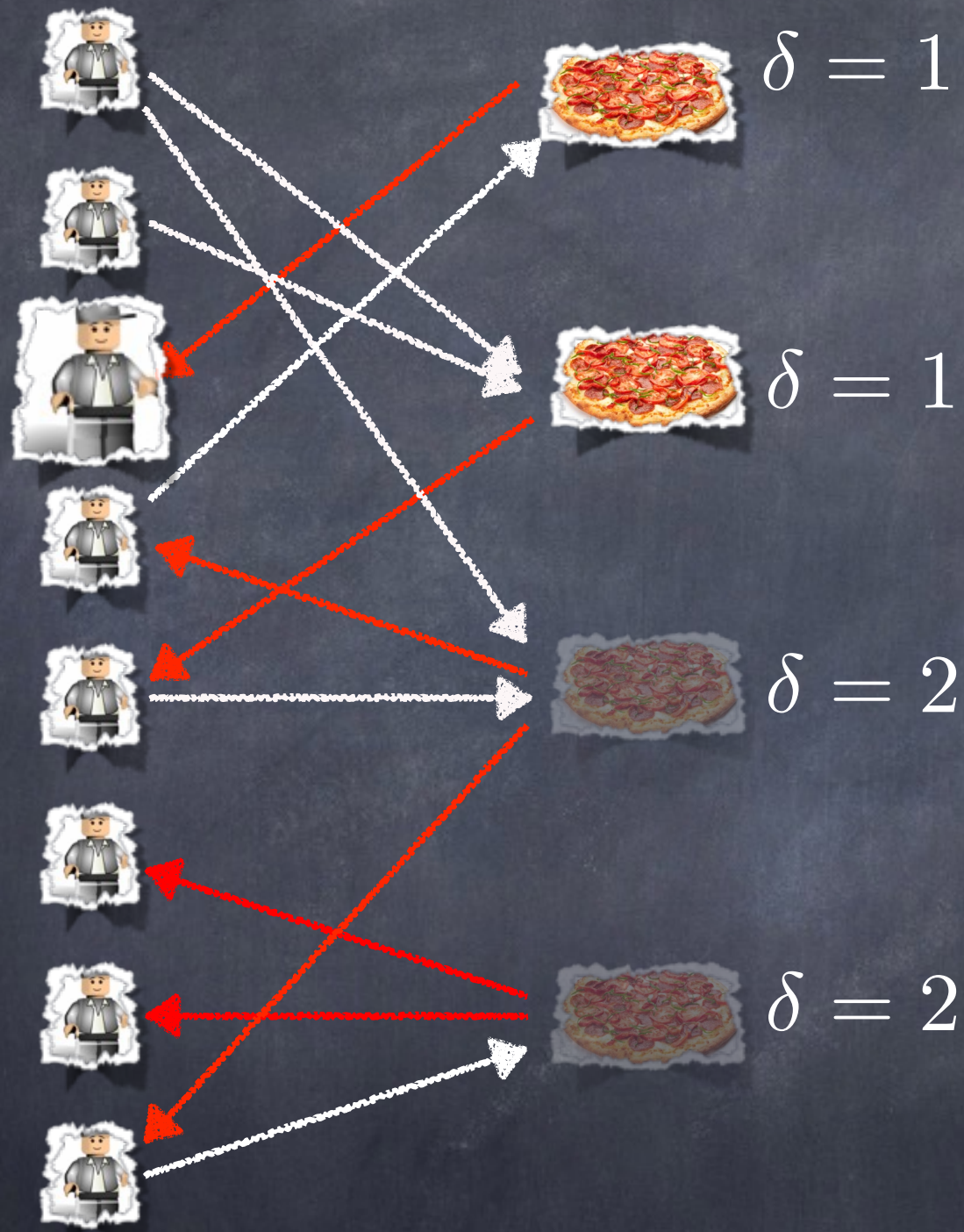
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

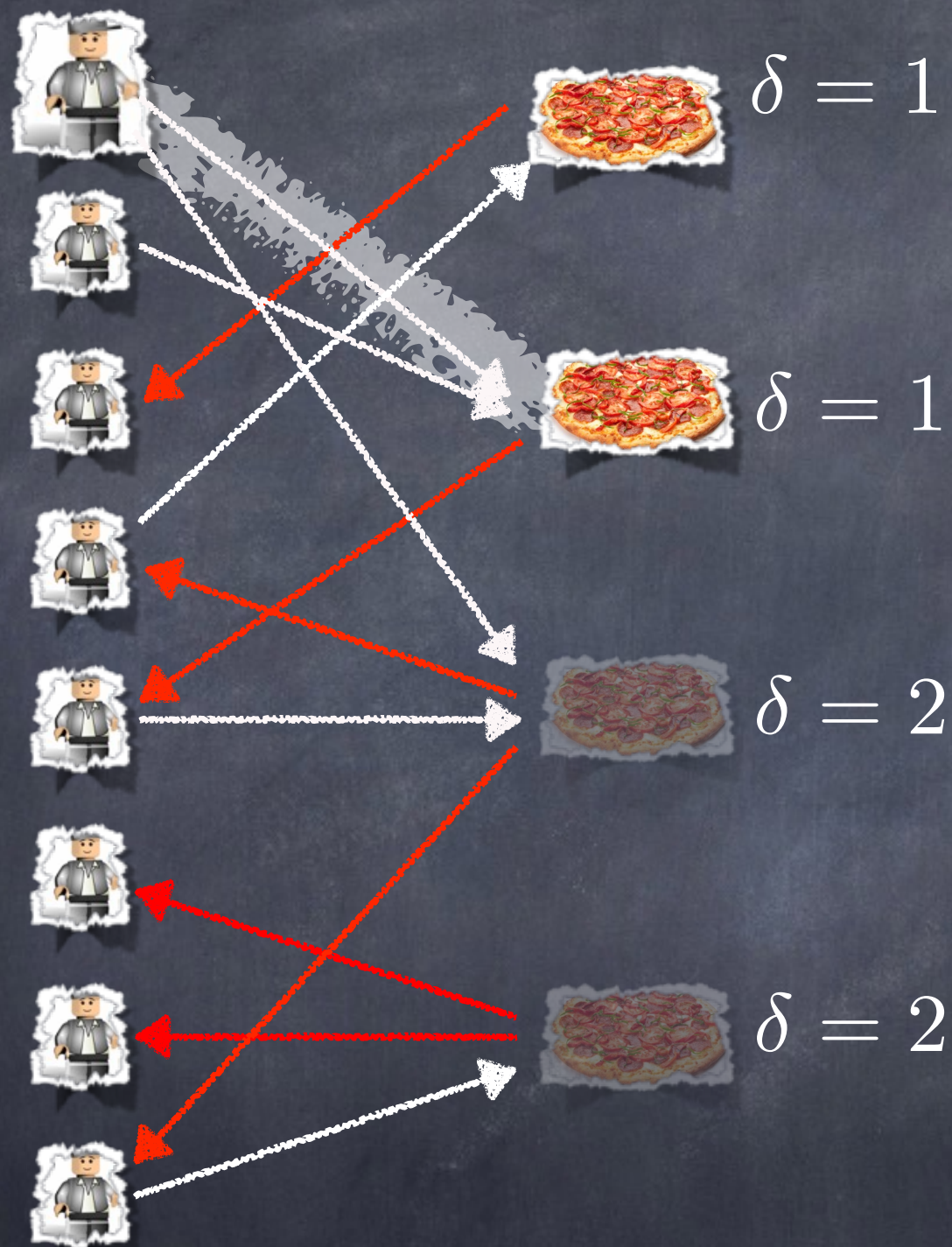
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

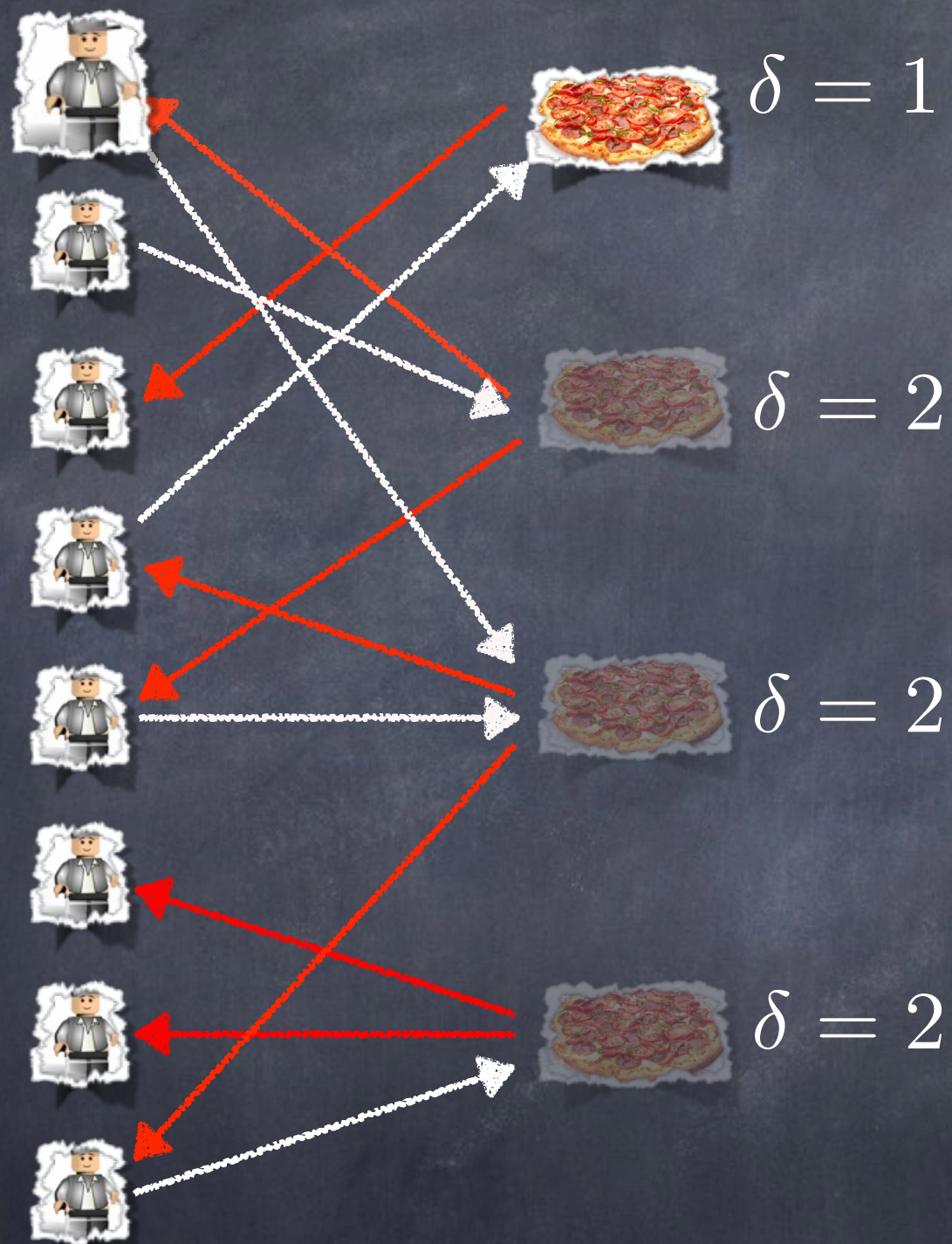
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

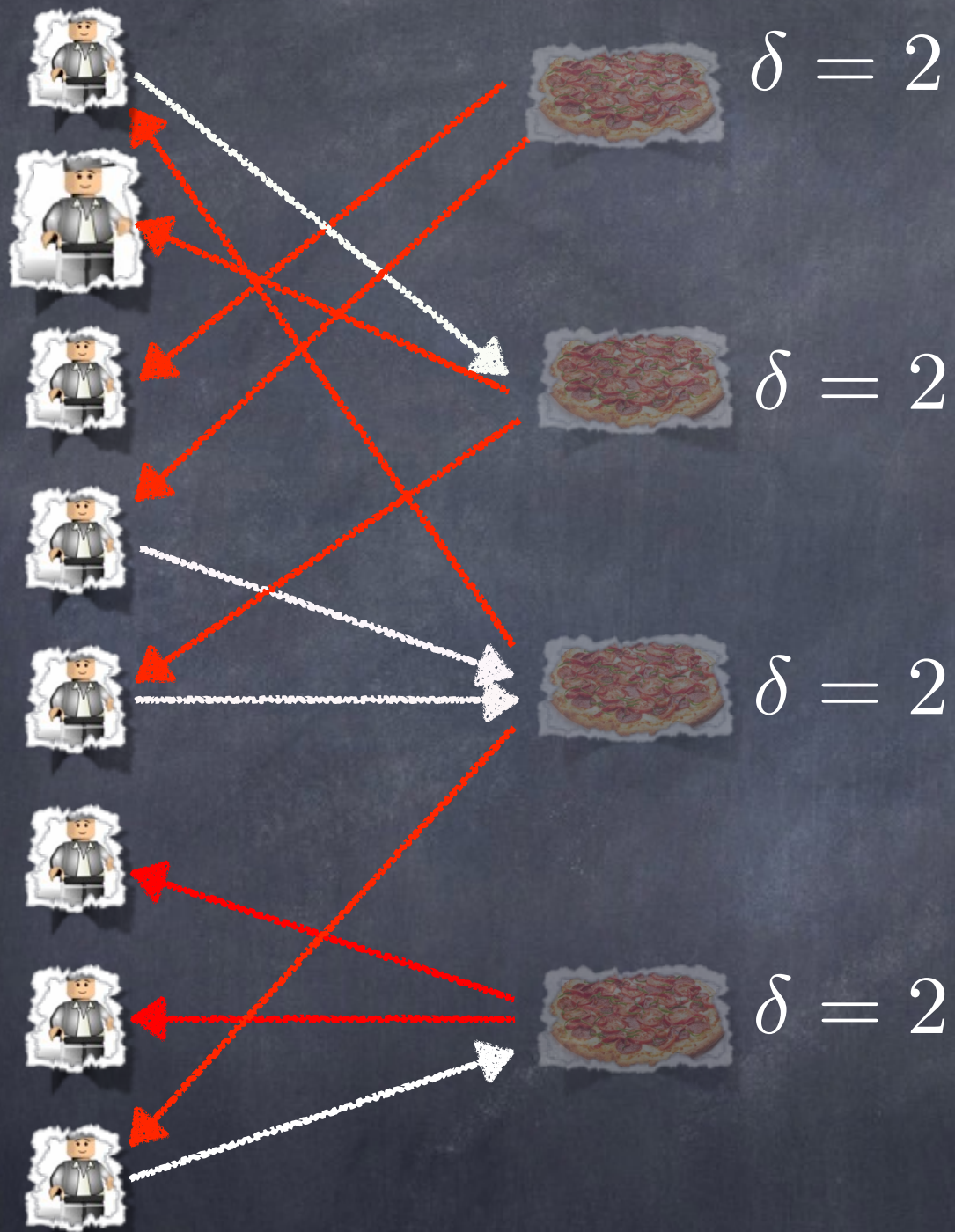
Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

Pizzabestellung



wie verbessern
wir E'

$n = 8, m = 4$ max. Größe: 2

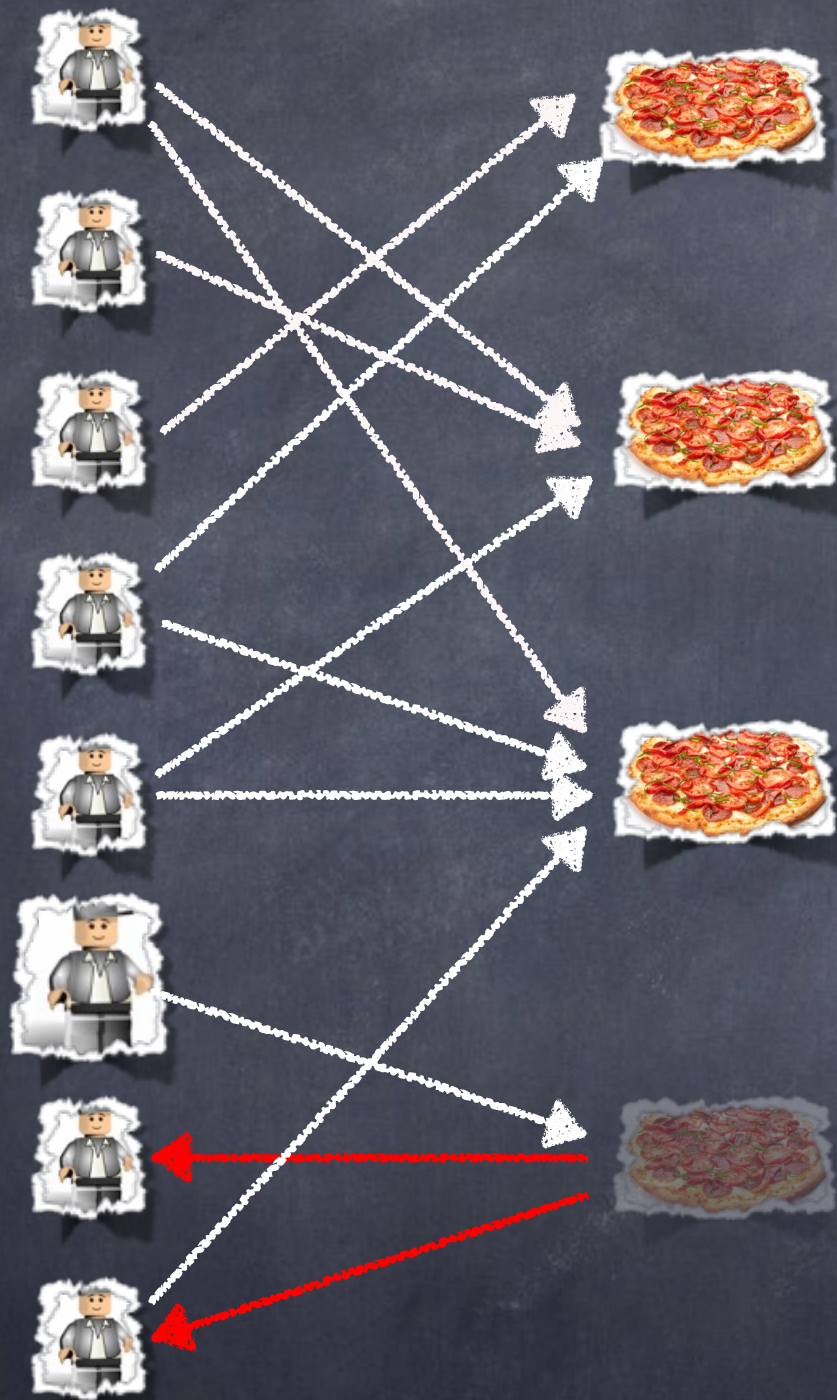
Pizzabestellung

wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“



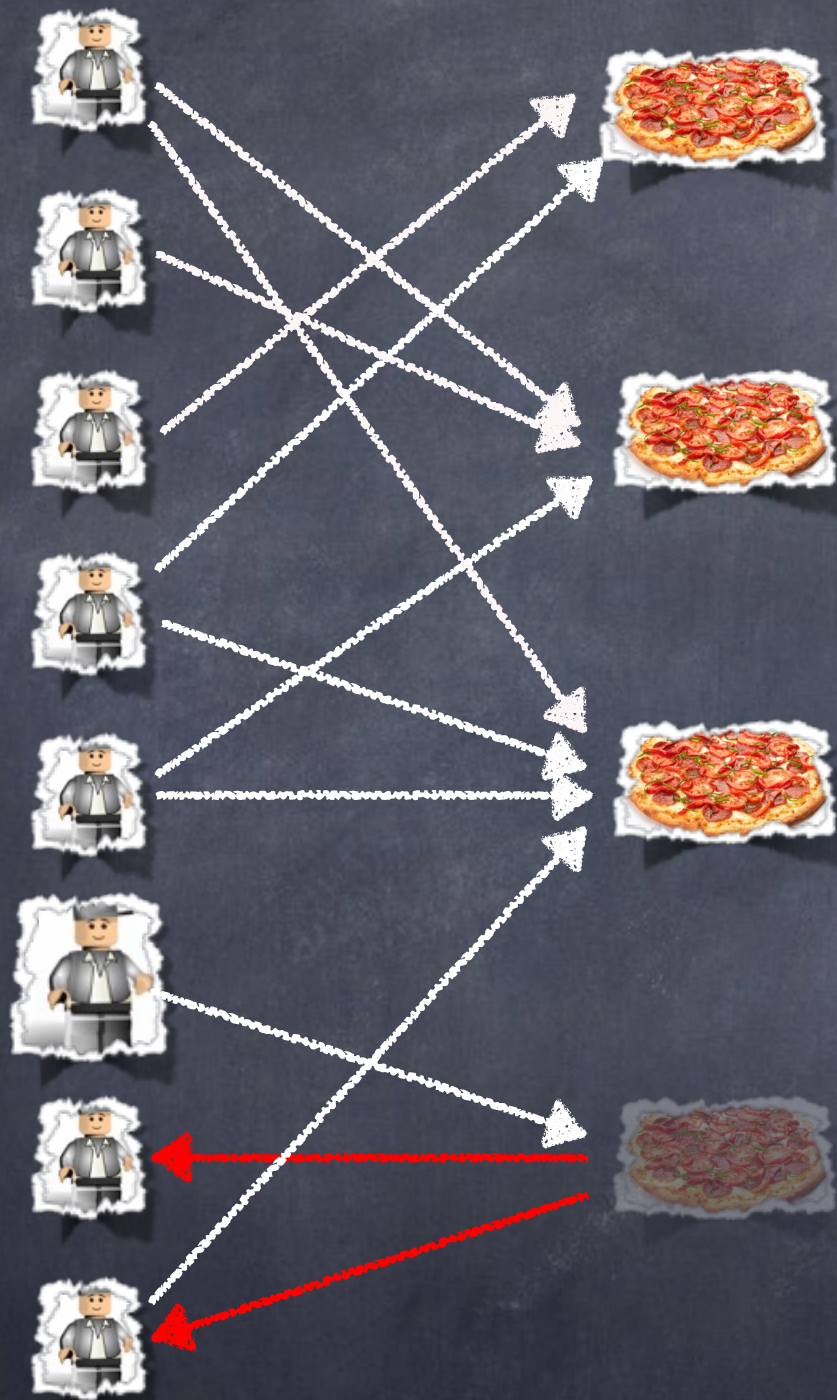
Pizzabestellung

wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“
 $\text{verbessere}(G, E')$



Pizzabestellung

wie verbessern
wir E'

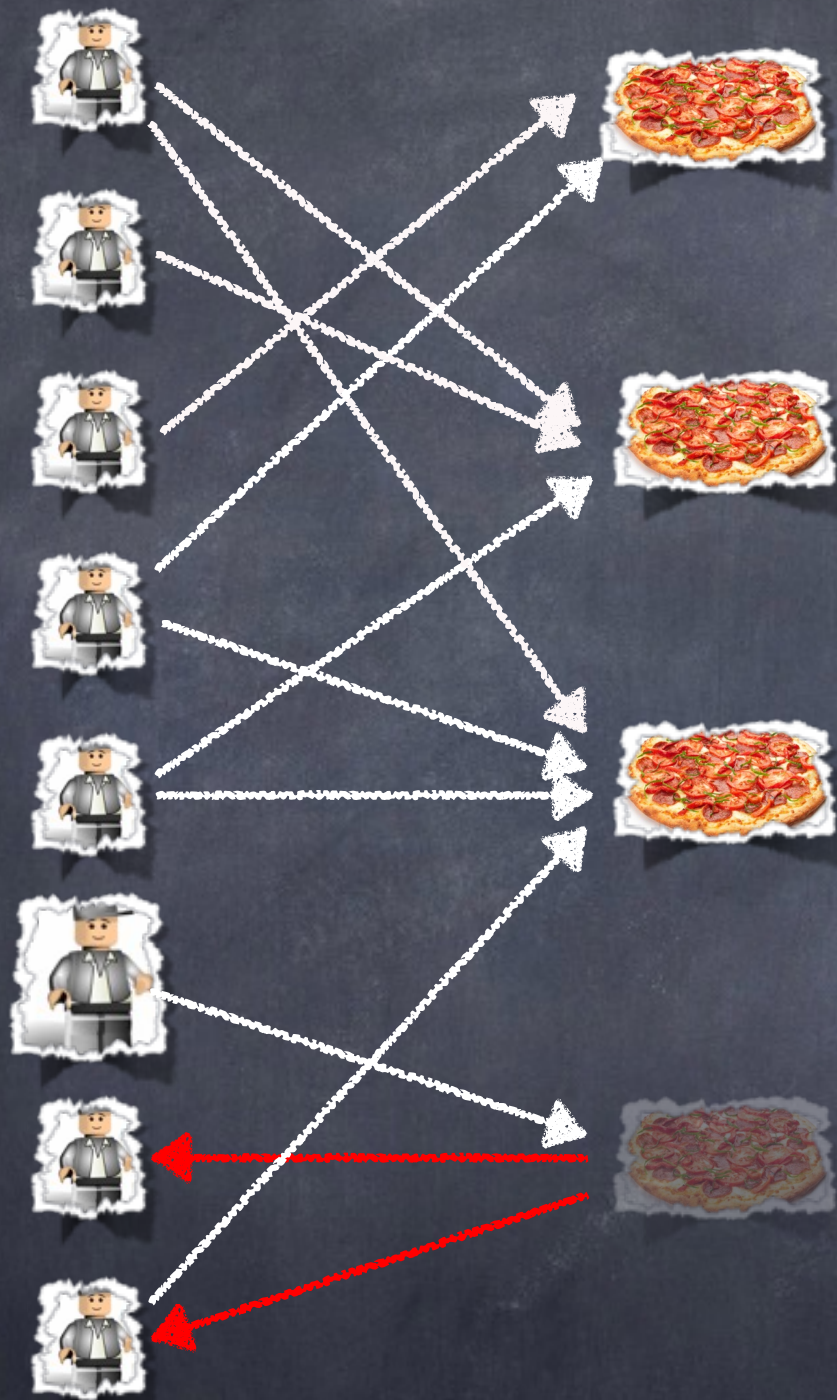
Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$



Pizzabestellung

wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

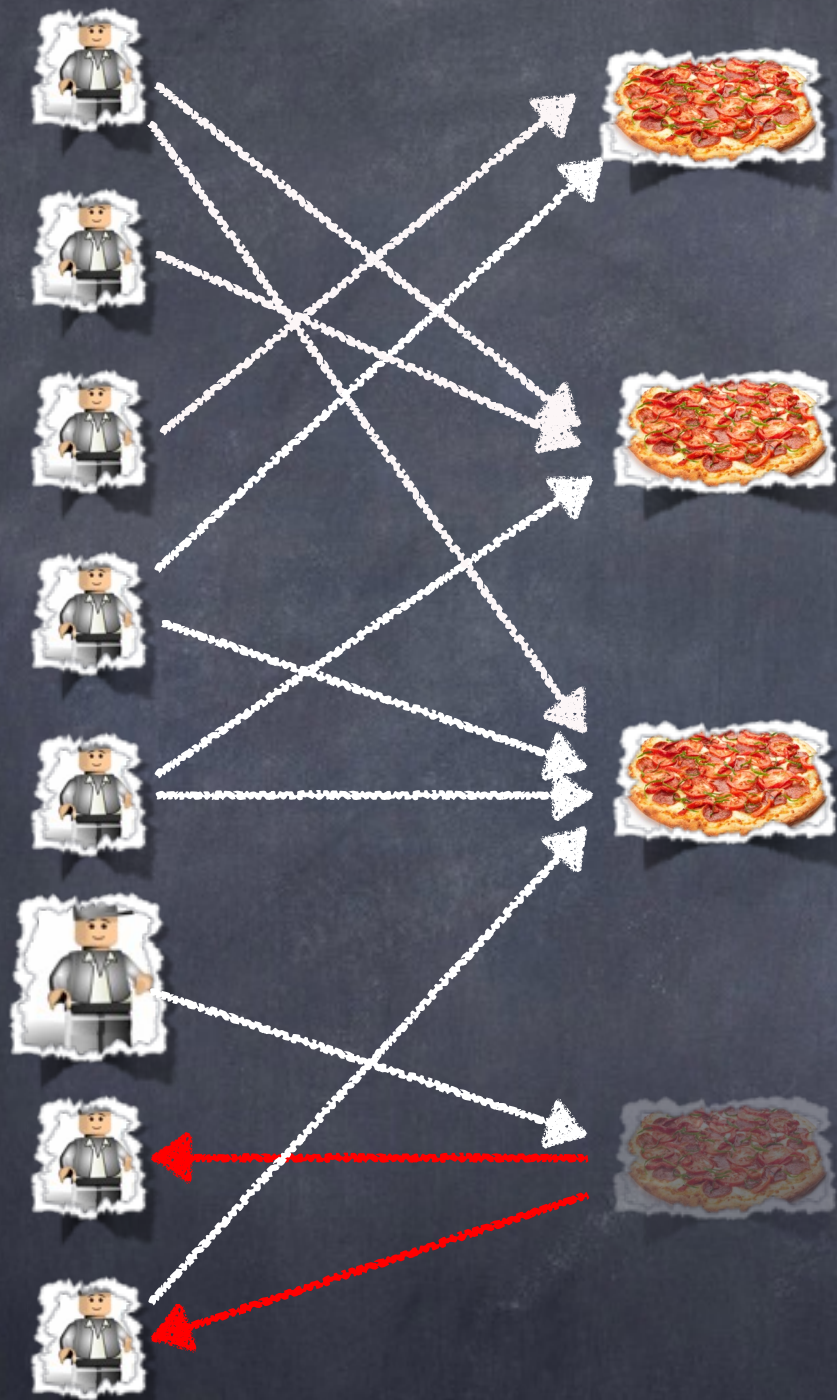
Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

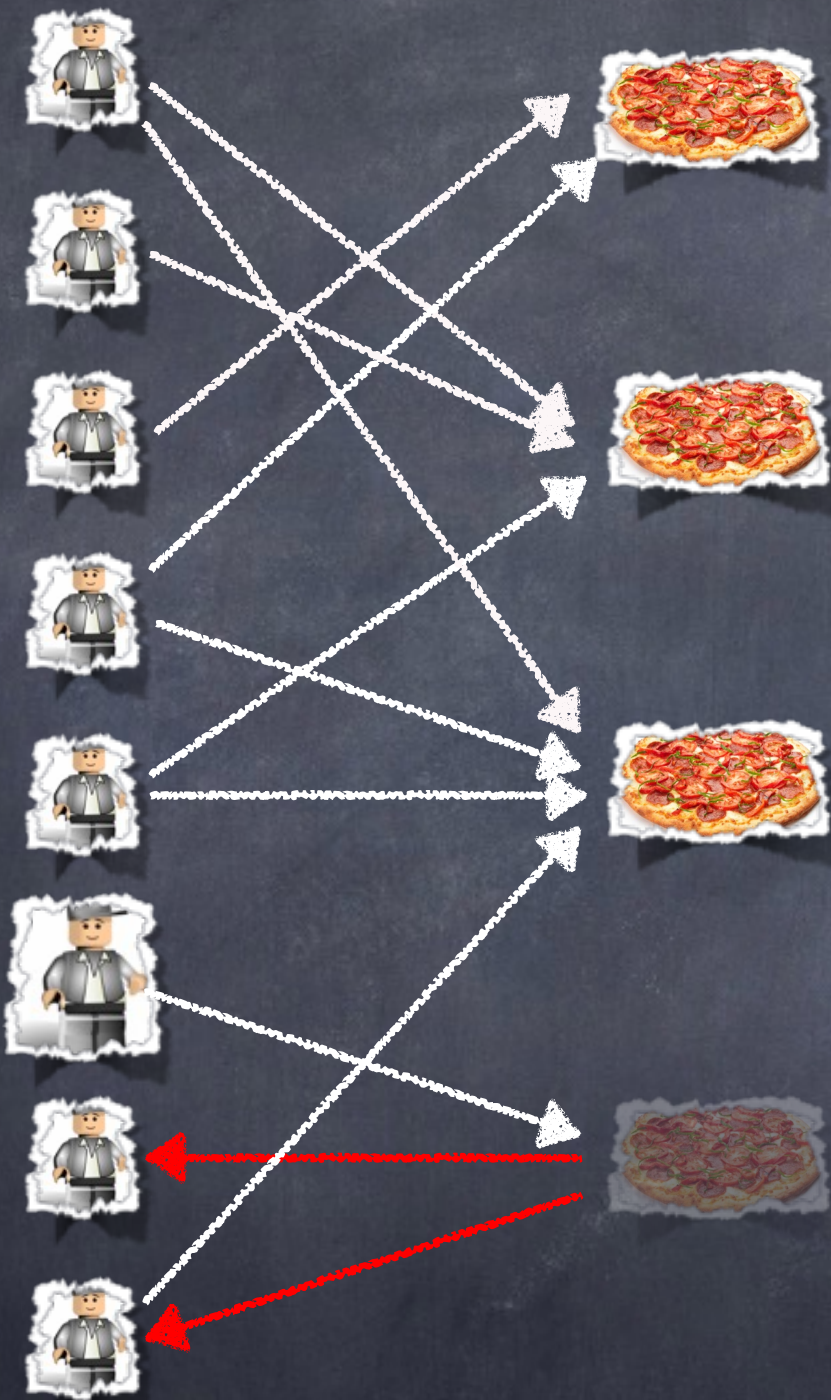
$$E'' = \emptyset \quad G'' = (V, E'')$$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$$E'' = E'' \cup \{(w, v)\}$$



Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

verbessere(G, E')

$$E'' = \emptyset \quad G'' = (V, E'')$$

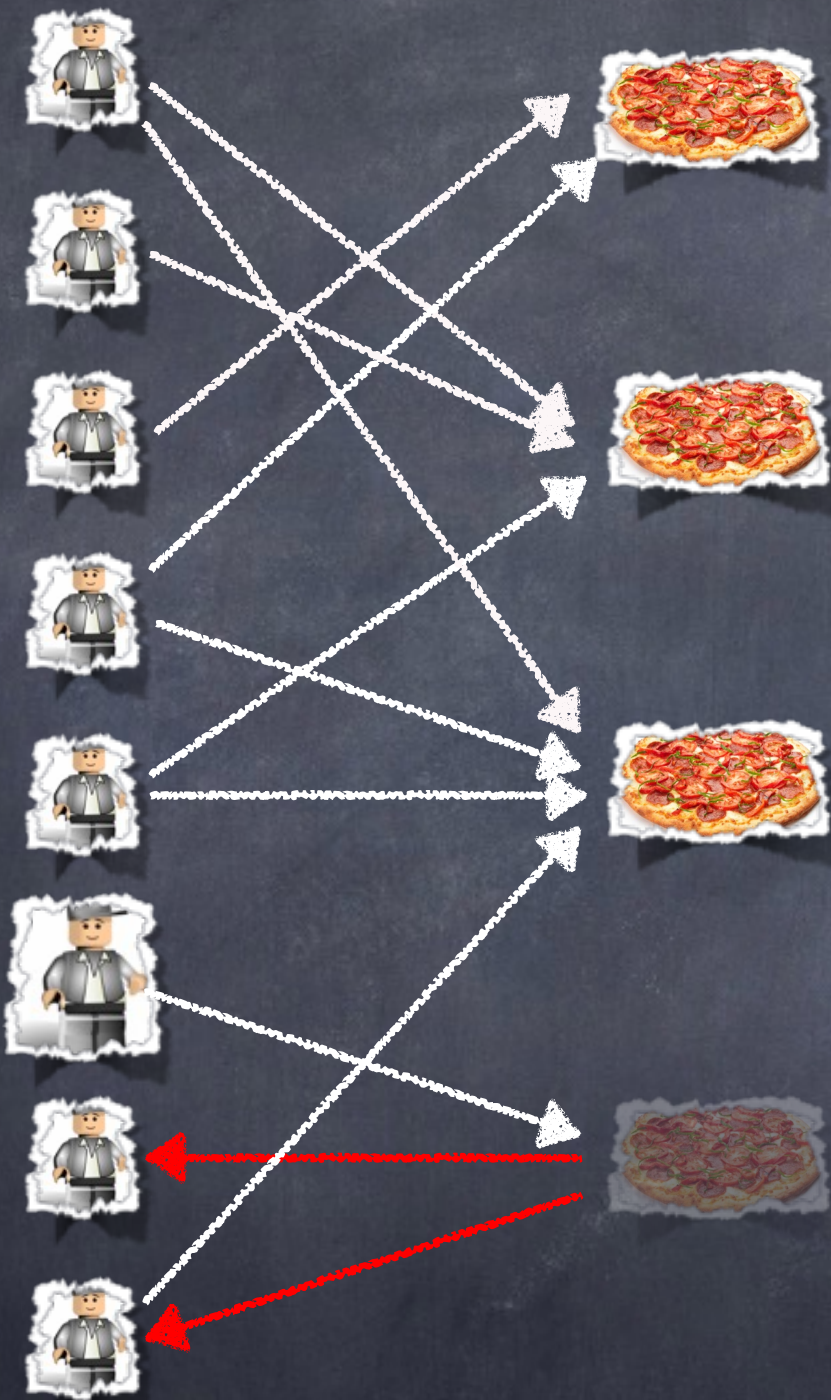
für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$$E'' = E'' \cup \{(w, v)\}$$

für alle $(v, w) \in E \setminus E' \subseteq V_1 \times V_2$

$$E'' = E'' \cup \{(v, w)\}$$

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$$E'' = \emptyset \quad G'' = (V, E'')$$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

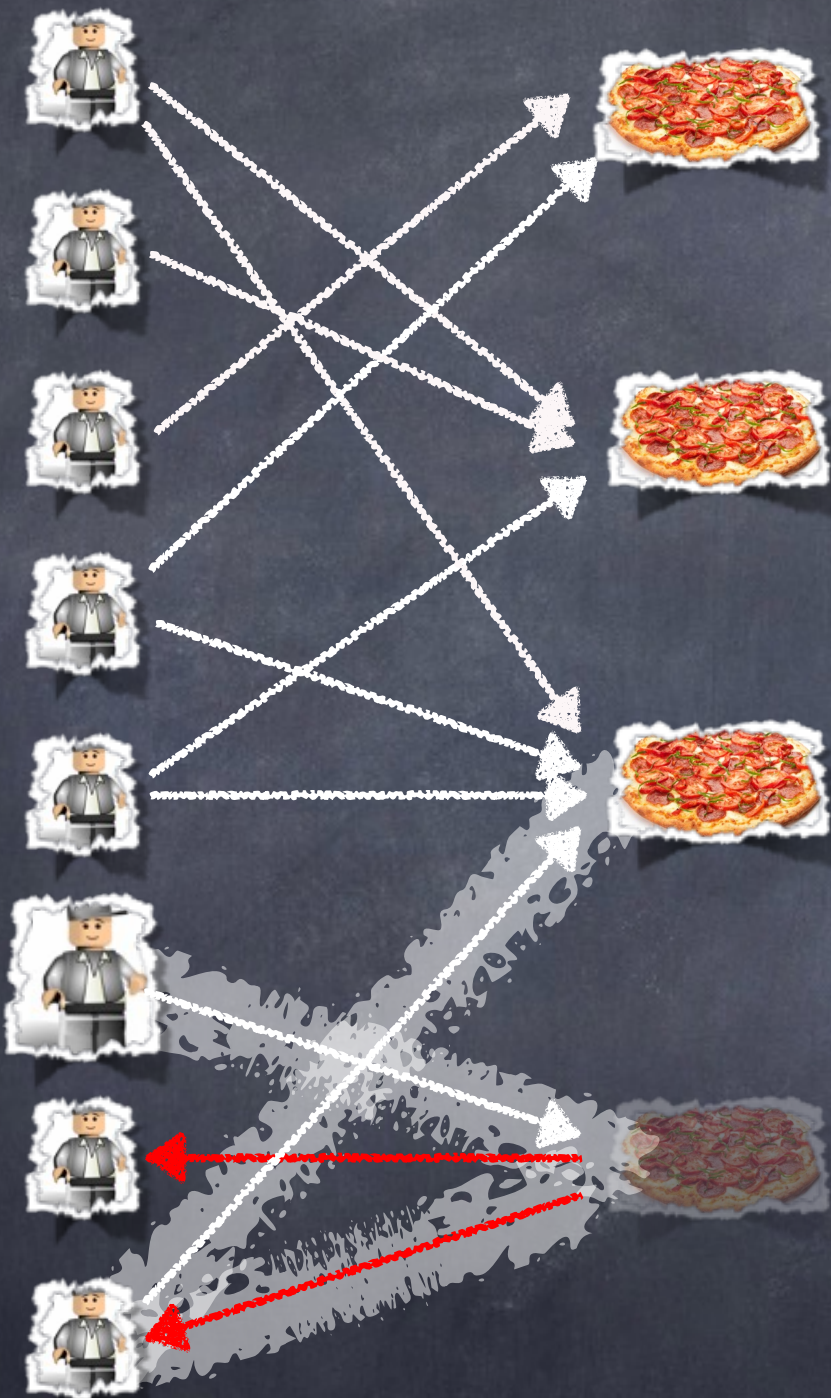
$$E'' = E'' \cup \{(w, v)\}$$

für alle $(v, w) \in E \setminus E' \subseteq V_1 \times V_2$

$$E'' = E'' \cup \{(v, w)\}$$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$$E'' = \emptyset \quad G'' = (V, E'')$$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$$E'' = E'' \cup \{(w, v)\}$$

für alle $(v, w) \in E \setminus E' \subseteq V_1 \times V_2$

$$E'' = E'' \cup \{(v, w)\}$$

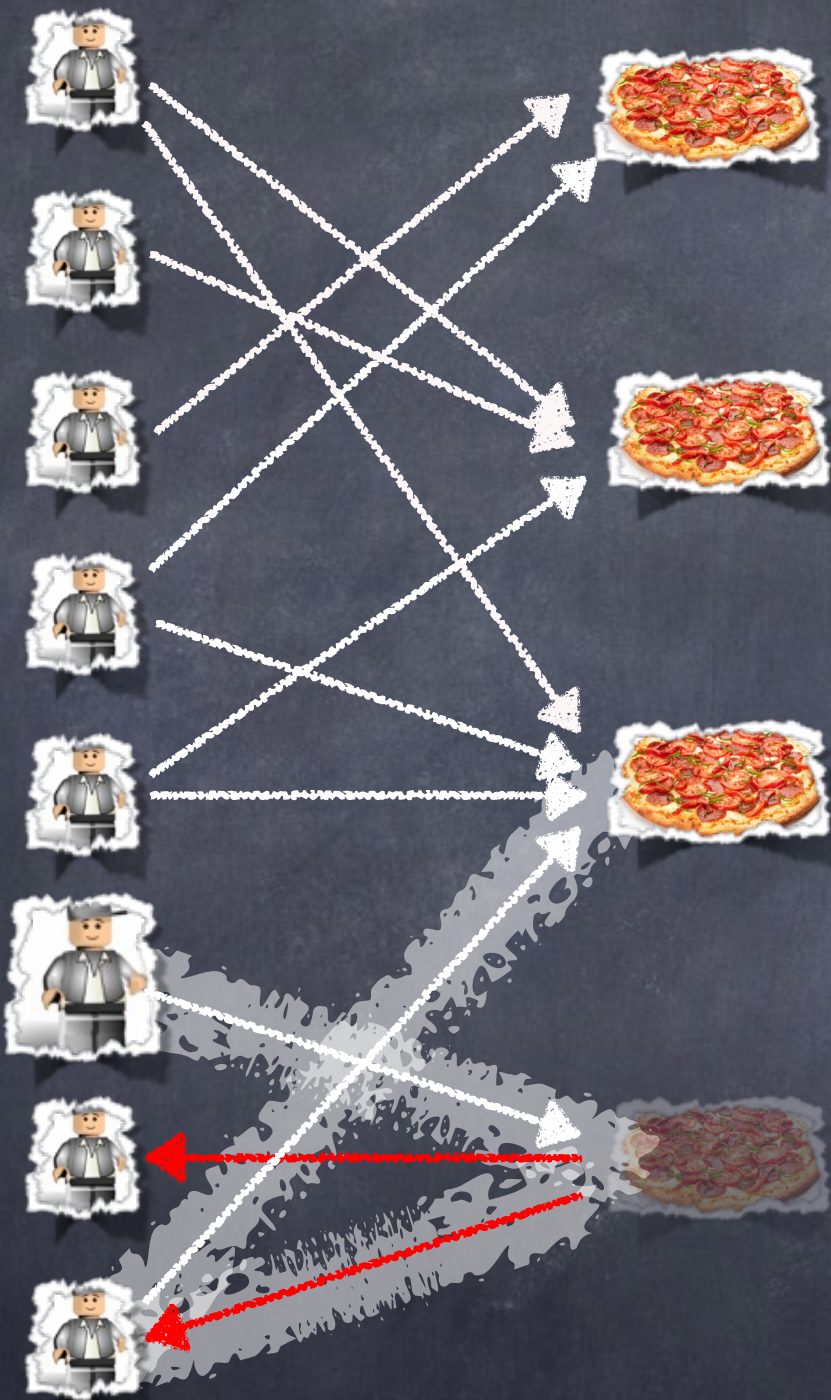
markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

Pfad π zu markiertem Knoten

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

Pfad π zu markiertem Knoten

für alle $(p, q) \in \pi$

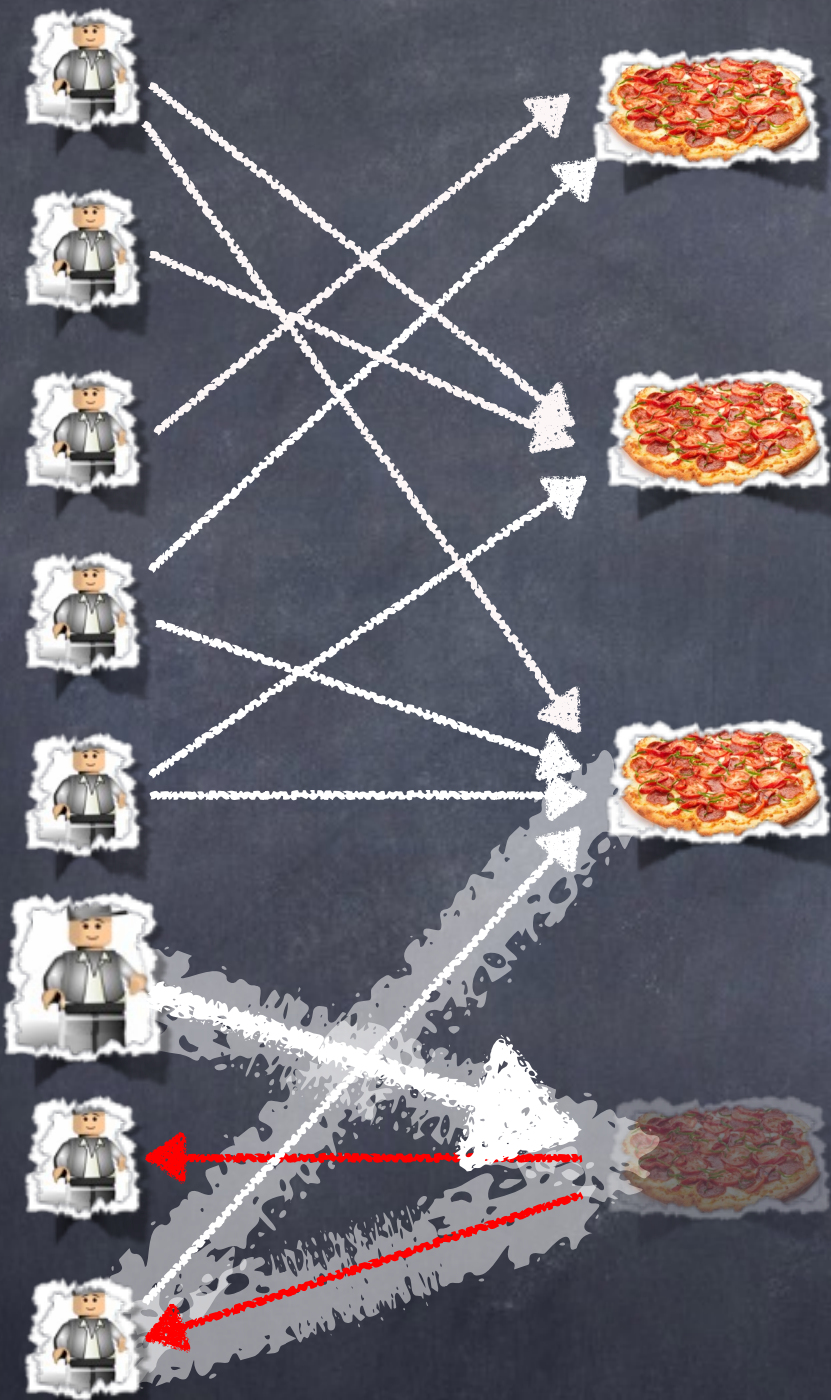
falls $(p, q) \in E'$ dann

$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

Pfad π zu markiertem Knoten

für alle $(p, q) \in \pi$

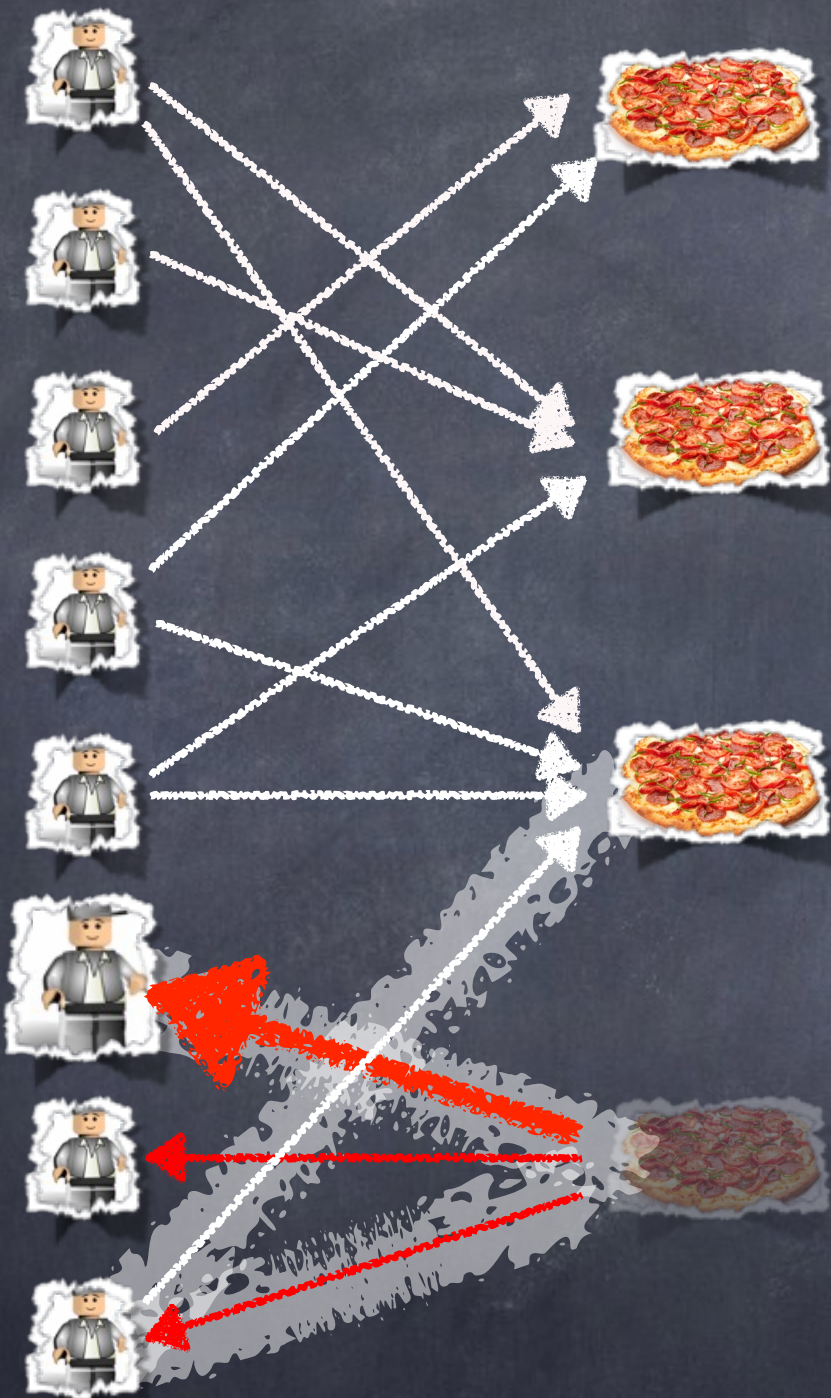
falls $(p, q) \in E'$ dann

$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subset V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

Pfad π zu markiertem Knoten

für alle $(p, q) \in \pi$

falls $(p, q) \in E'$ dann

$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

wie verbessern
wir E'

ALLg.

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

verbessere(G, E')

$$E'' = \emptyset \quad G'' = (V, E'')$$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$$E'' = E'' \cup \{(w, v)\}$$

für alle $(v, w) \in E \setminus E' \subset V_1 \times V_2$

$$E'' = E'' \cup \{(v, w)\}$$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

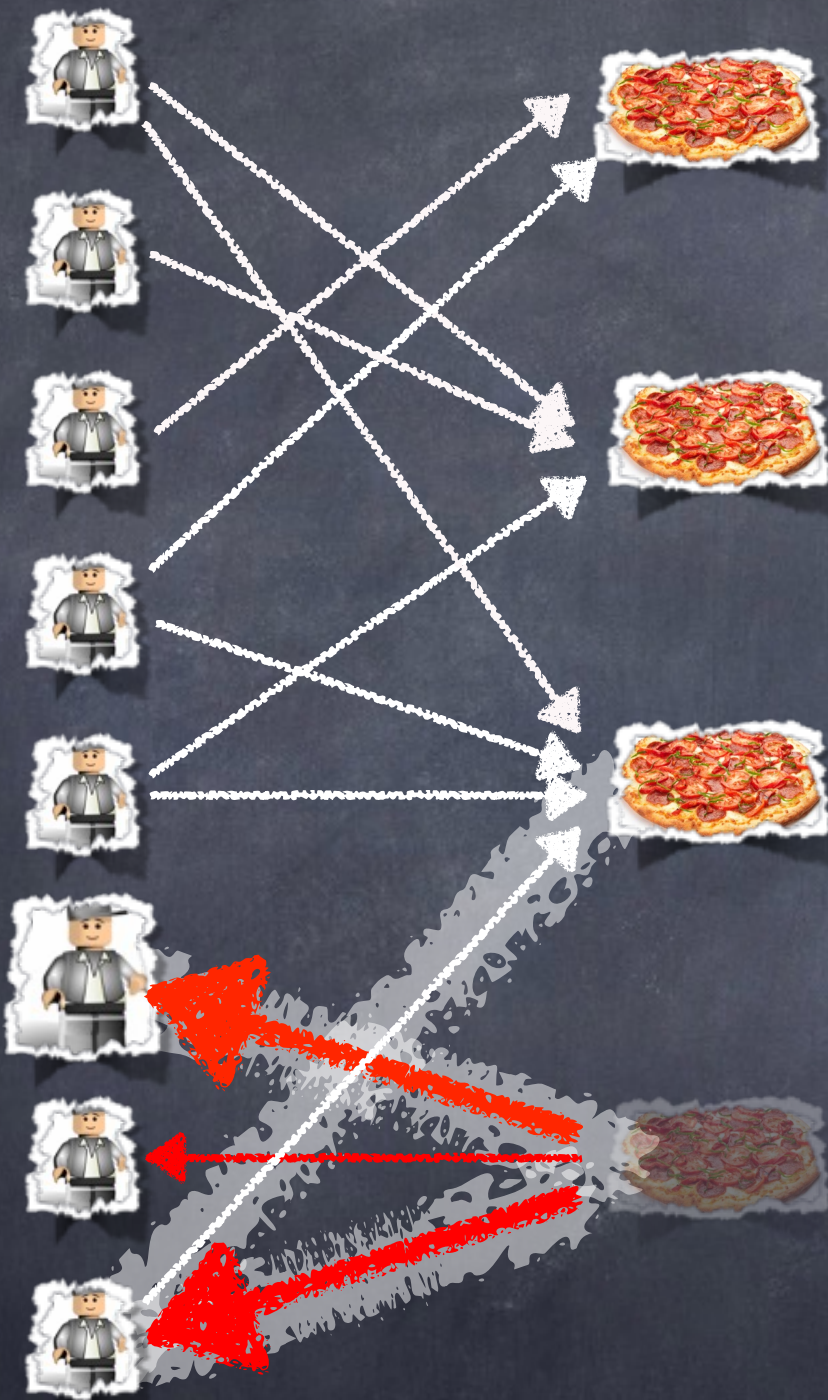
Pfad π zu markiertem Knoten

für alle $(p, q) \in \pi$

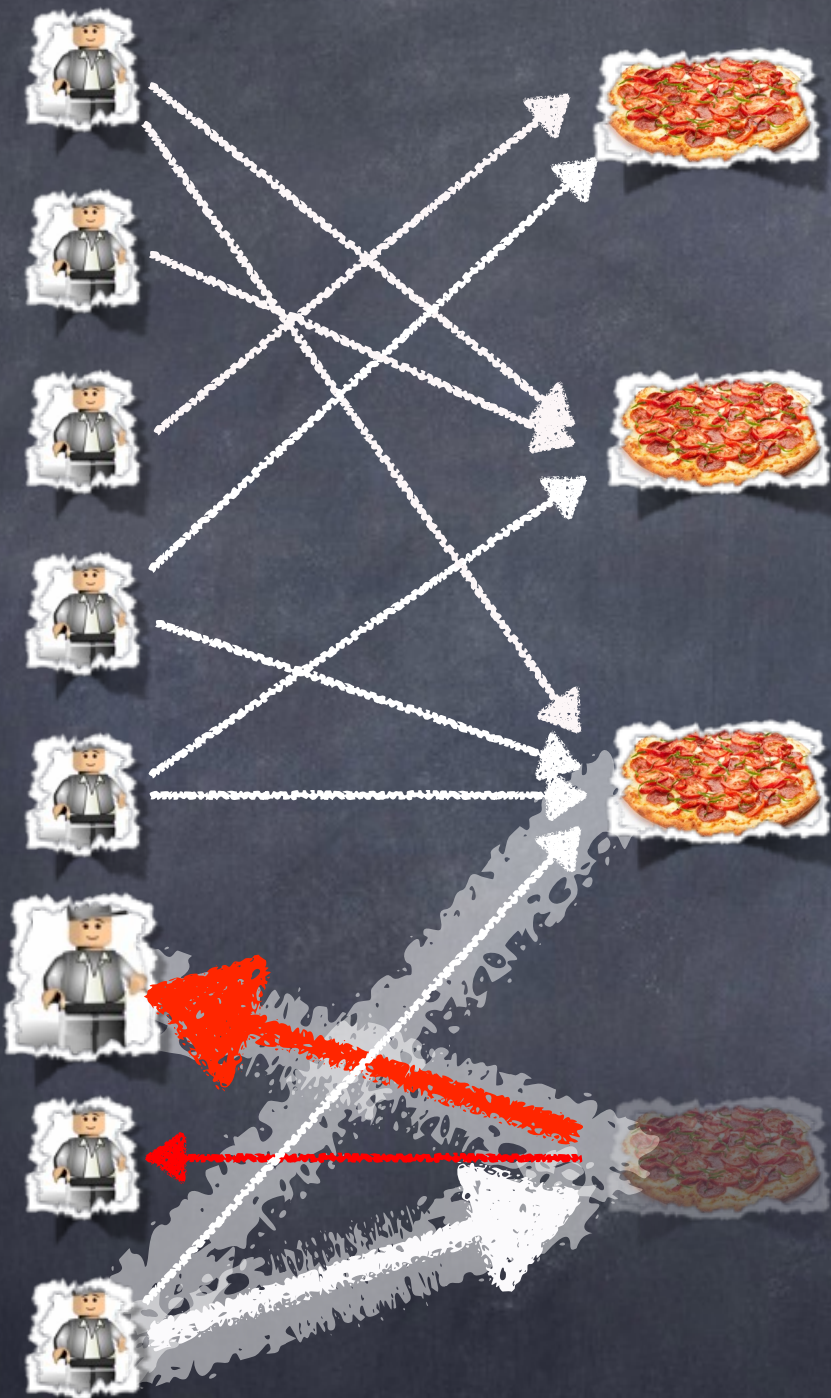
falls $(p, q) \in E'$ dann

$$E' = E' \setminus \{(p, q)\}$$

falls $(p, q) \notin E'$

$$E' = E' \cup \{(p, q)\}$$


Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subset V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

Pfad π zu markiertem Knoten

für alle $(p, q) \in \pi$

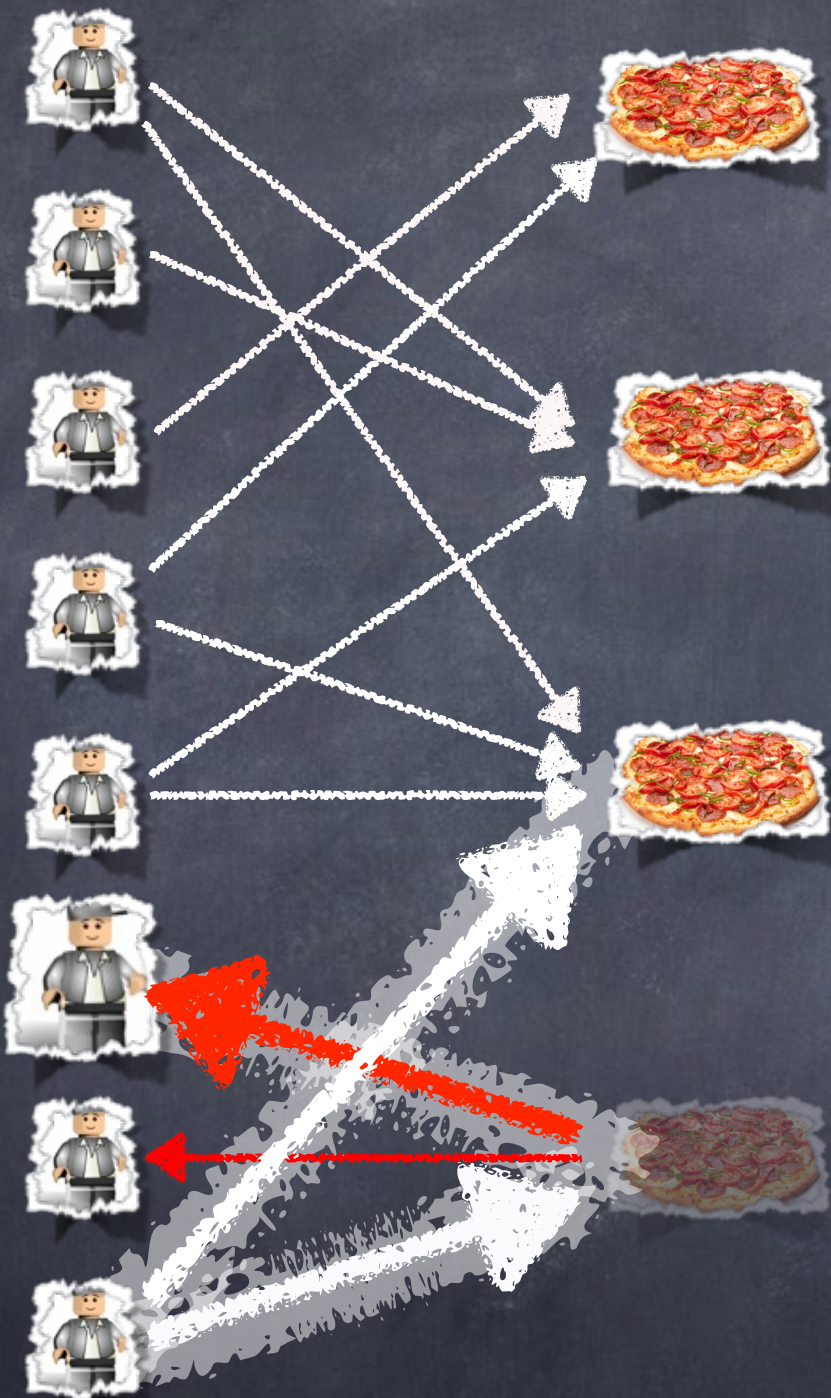
falls $(p, q) \in E'$ dann

$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subset V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

Pfad π zu markiertem Knoten

für alle $(p, q) \in \pi$

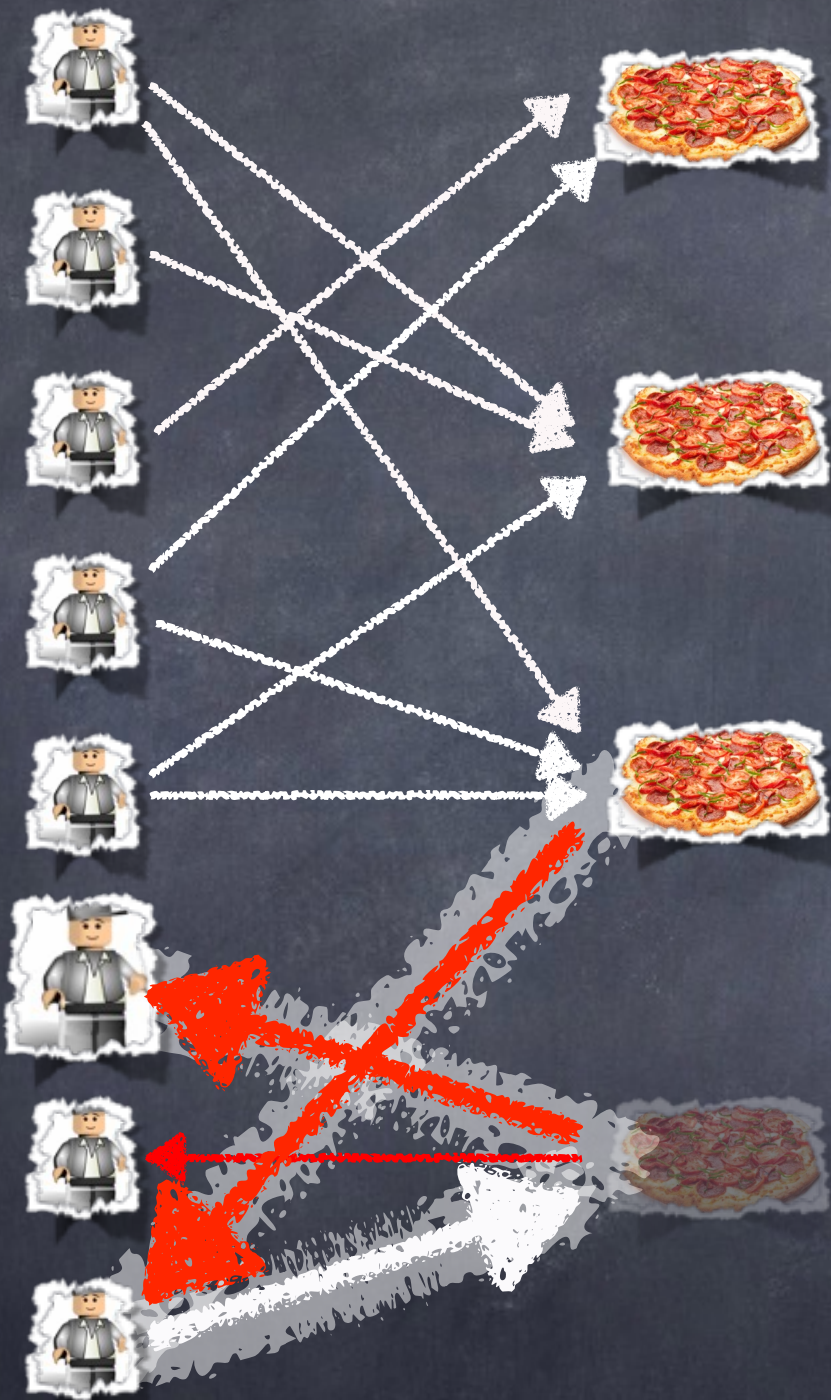
falls $(p, q) \in E'$ dann

$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subset V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

Pfad π zu markiertem Knoten

für alle $(p, q) \in \pi$

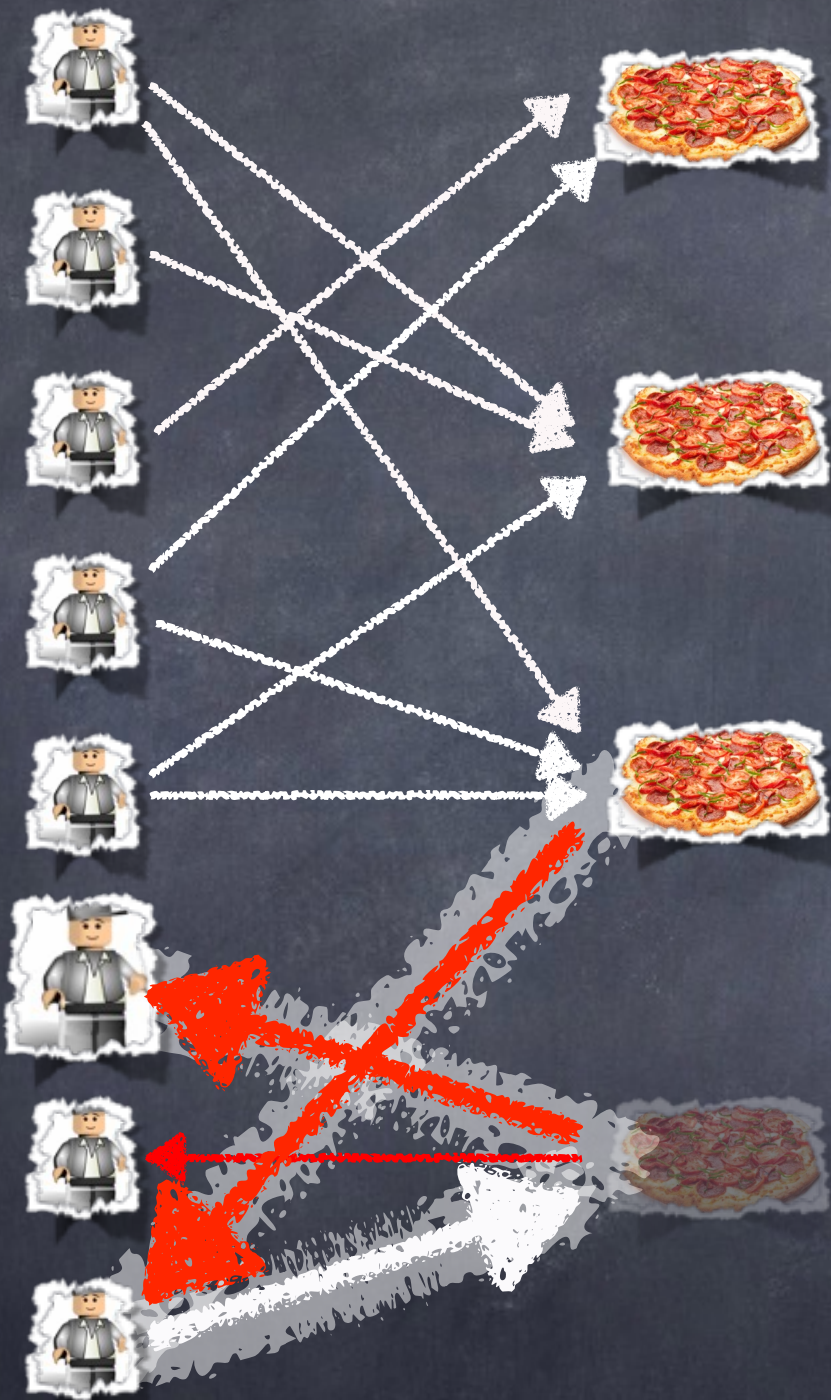
falls $(p, q) \in E'$ dann

$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

verbessere(G, E')

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subset V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

BFS(v) oder DFS(v)
Pfad zu markiertem Knoten

für alle $(p, q) \in \pi$

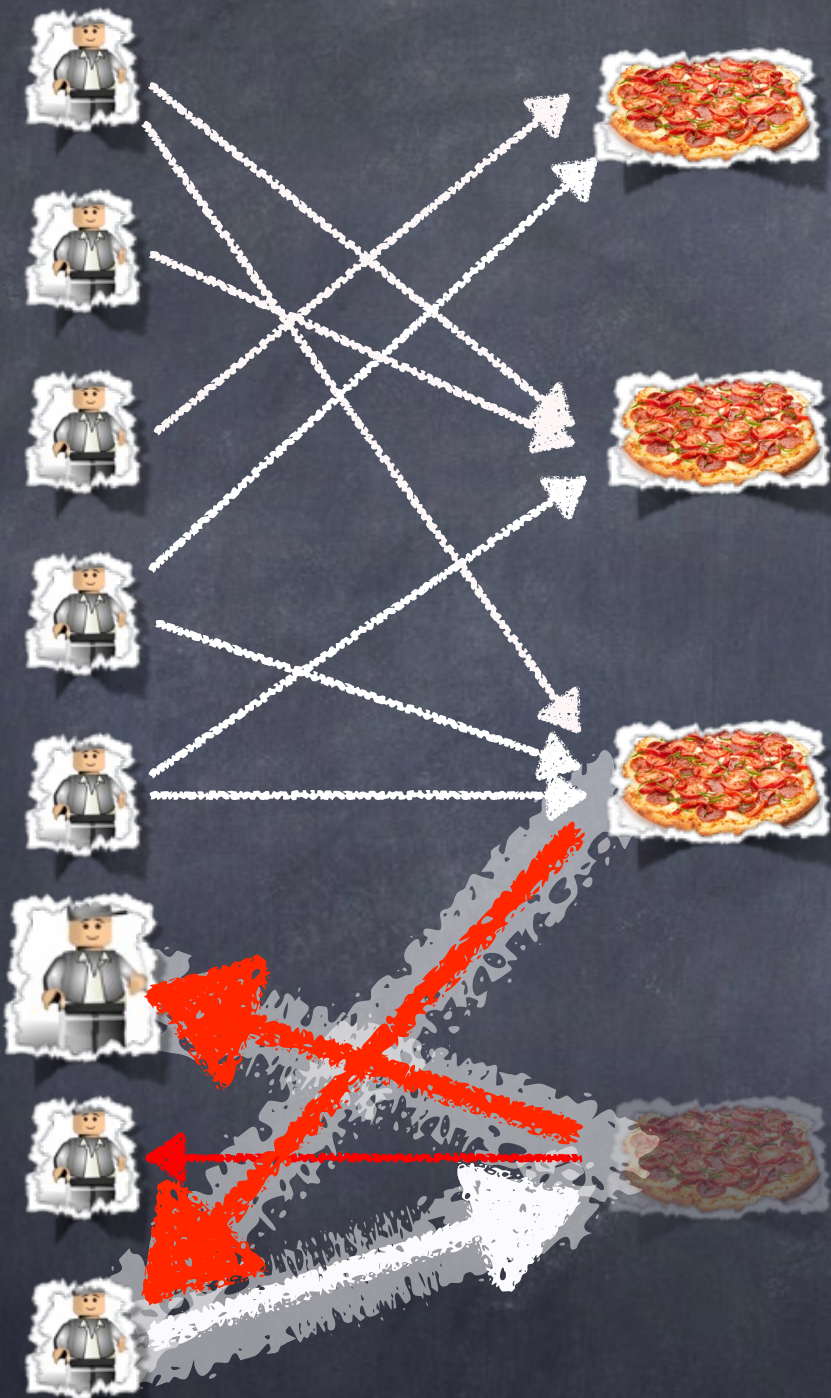
falls $(p, q) \in E'$ dann

$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subset V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

BFS(v) oder DFS(v)
Pfad zu markierten Knoten

für alle $(p, q) \in \pi$

falls $(p, q) \in E'$ dann

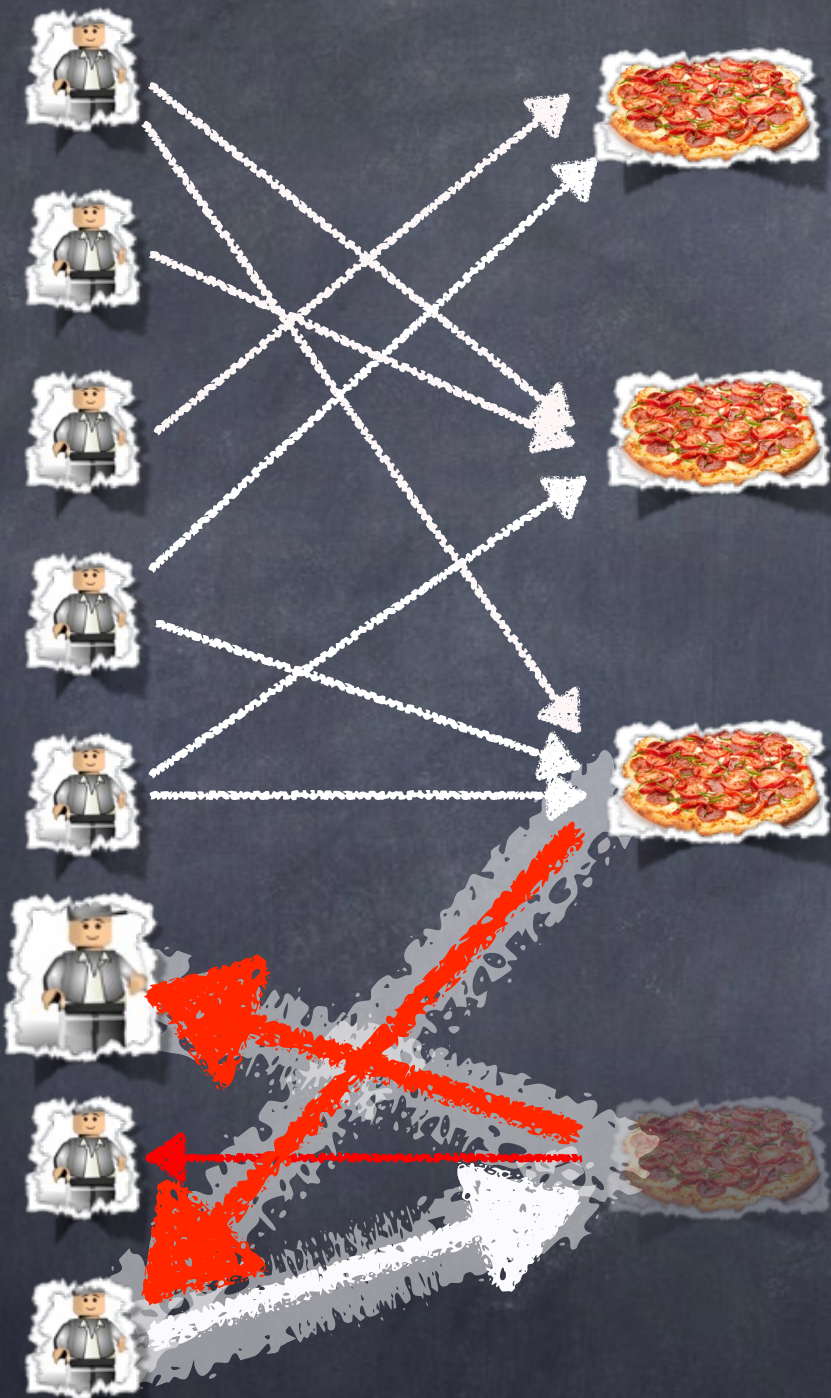
$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

BFS und DFS suchen doch
nur nach Knoten und nicht
nach Pfaden?!

Pizzabestellung



wie verbessern
wir E'

Allg.:

Geg.: $G = (V, E)$ und E'

Ges.: „verbessernder Pfad“

$\text{verbessere}(G, E')$

$E'' = \emptyset \quad G'' = (V, E'')$

für alle $(v, w) = e' \in E' \subseteq V_1 \times V_2$

$E'' = E'' \cup \{(w, v)\}$

für alle $(v, w) \in E \setminus E' \subset V_1 \times V_2$

$E'' = E'' \cup \{(v, w)\}$

markiere alle $w \in V_2$ mit $\delta(w) < 2$

für alle $v \in V_1$

suche ausgehend von v

BFS(v) oder DFS(v)
Pfad zu markierten Knoten

für alle $(p, q) \in \pi$

falls $(p, q) \in E'$ dann

$E' = E' \setminus \{(p, q)\}$

falls $(p, q) \notin E'$

$E' = E' \cup \{(p, q)\}$

BFS und DFS suchen doch
nur nach Knoten und nicht
nach Pfaden?!

Nicht
ganz!

Pizzabestellung

Wiederholung

BFS

Pizzabestellung

Wiederholung

BFS

BFS = breadth-first search

Pizzabestellung

Wiederholung

BFS

BFS = breadth-first search
(Breitensuche)

Pizzabestellung

Wiederholung

BFS

BFS = breadth-first search
(Breitensuche)

verw. Datenstruktur:

Pizzabestellung

Wiederholung

BFS

BFS = breadth-first search
(Breitensuche)

verw. Datenstruktur:

Queue (Warteschlange)

Pizzabestellung

Wiederholung

BFS

BFS = breadth-first search
(Breitensuche)

verw. Datenstruktur:

Queue (Warteschlange)

first-in-first-out (fifo)

Pizzabestellung

Wiederholung

BFS

BFS = breadth-first search
(Breitensuche)

verw. Datenstruktur:

Queue (Warteschlange)

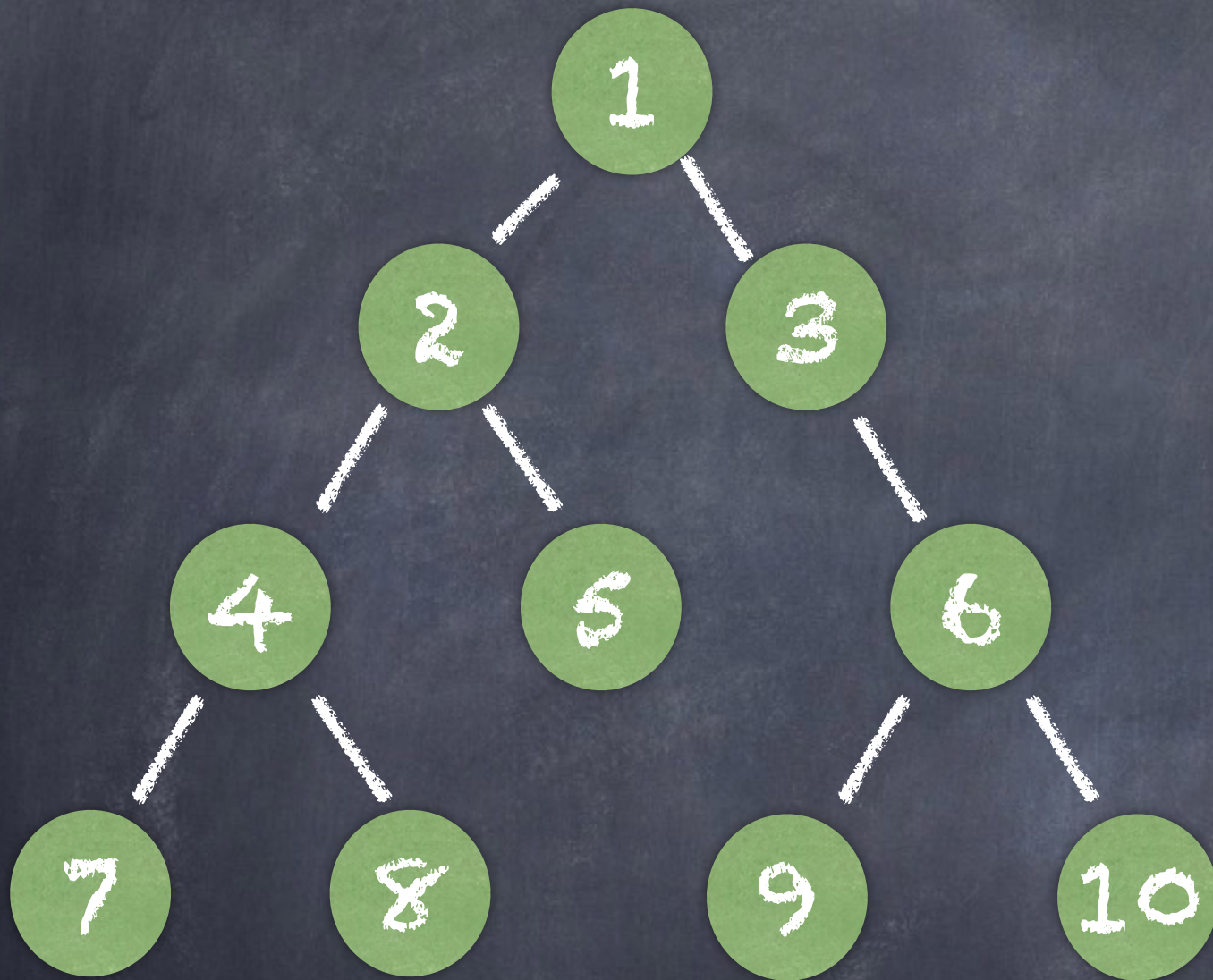
first-in-first-out (fifo)



Pizzabestellung

Wiederholung

BFS



BFS = breadth-first search
(Breitensuche)

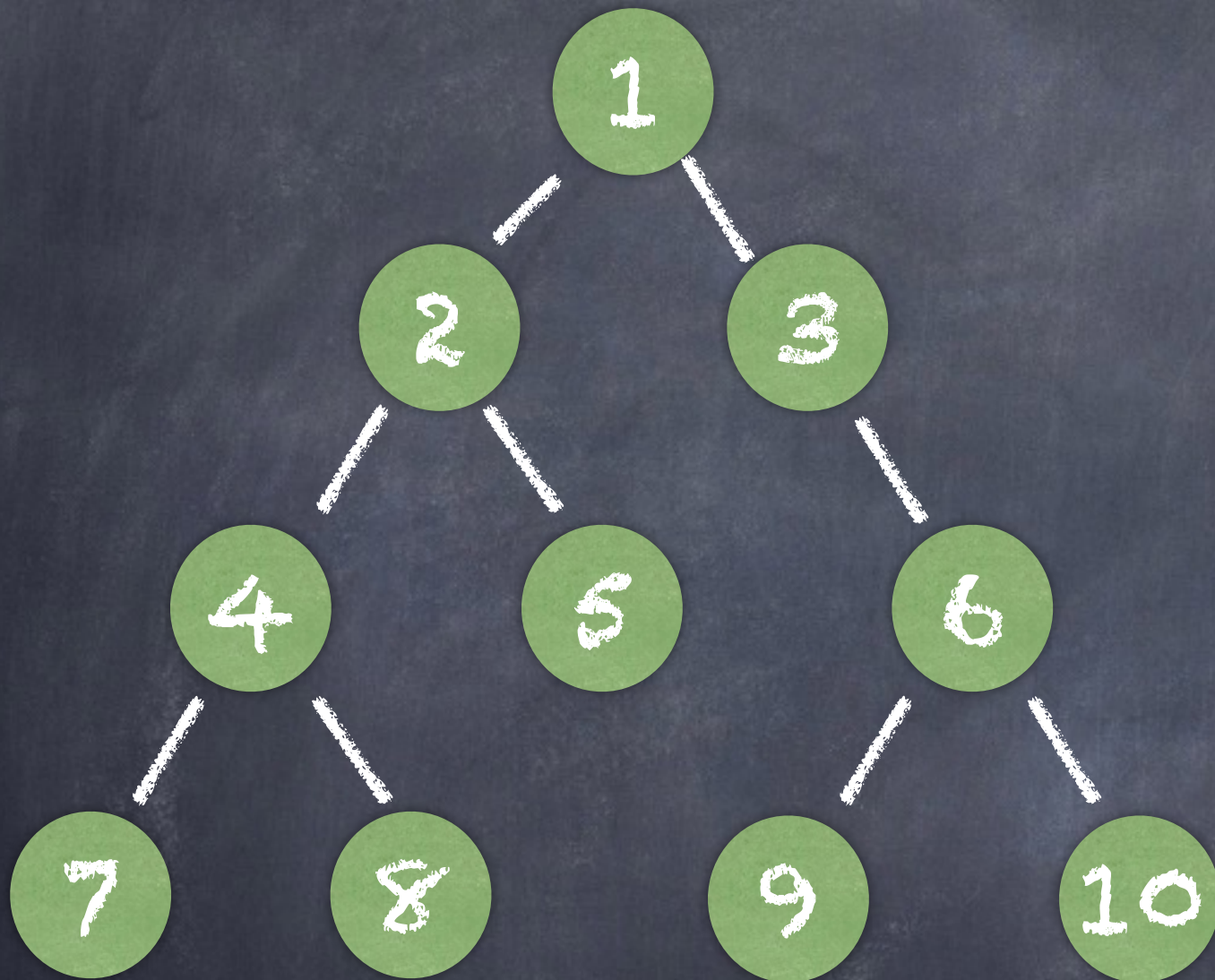
verw. Datenstruktur:

Queue (Warteschlange)
first-in-first-out (fifo)



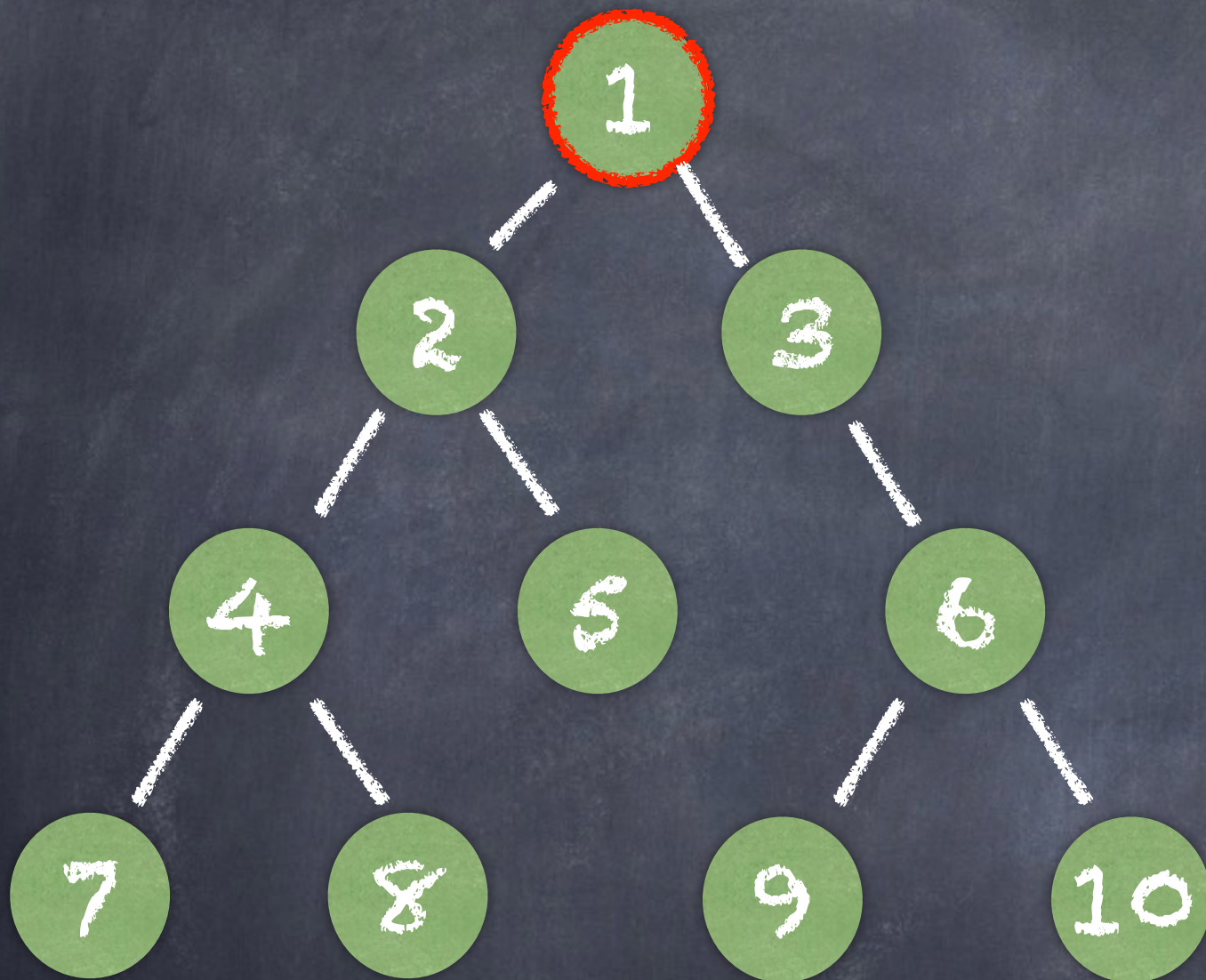
Pizzabestellung

Wiederholung
BFS



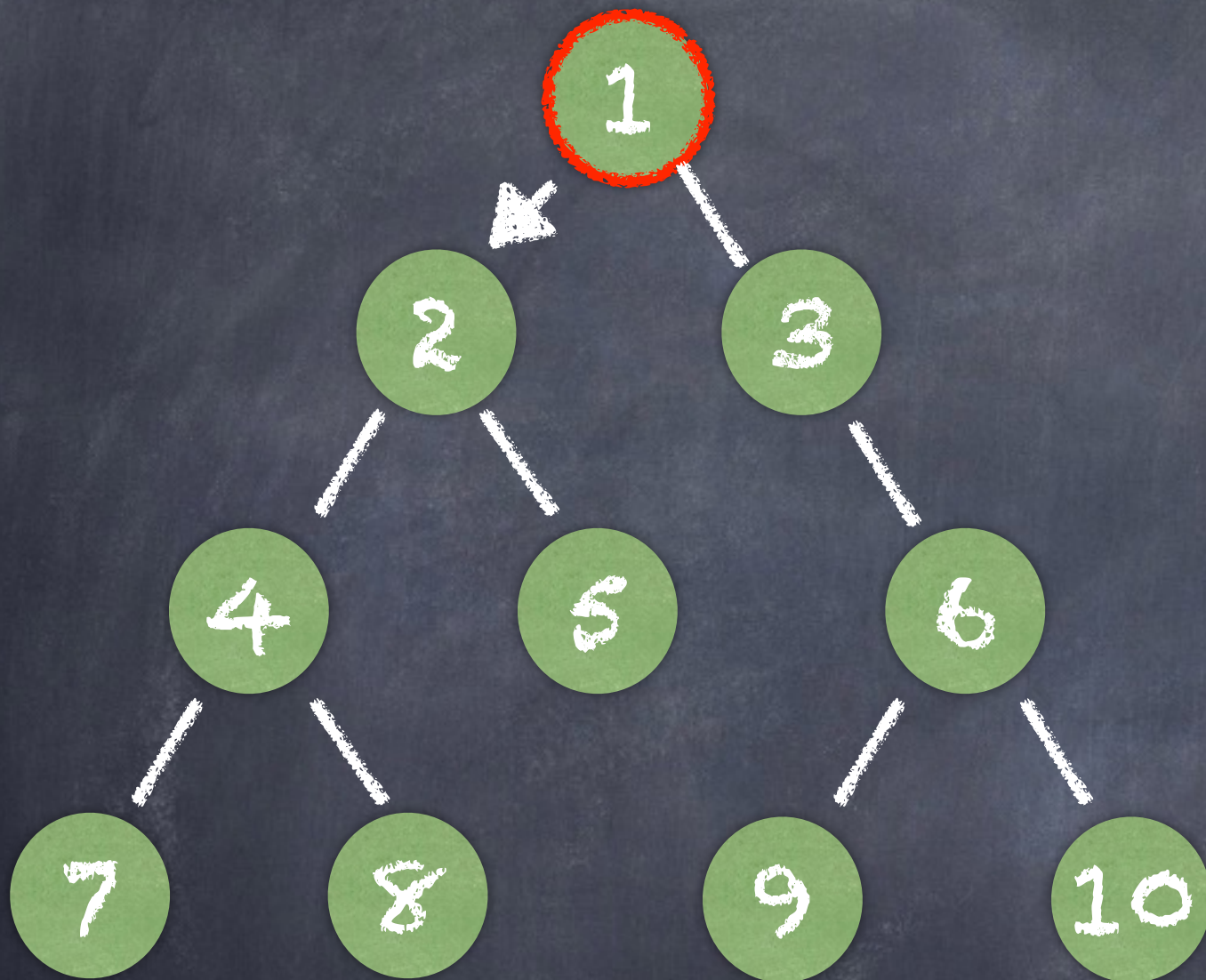
Pizzabestellung

Wiederholung
BFS



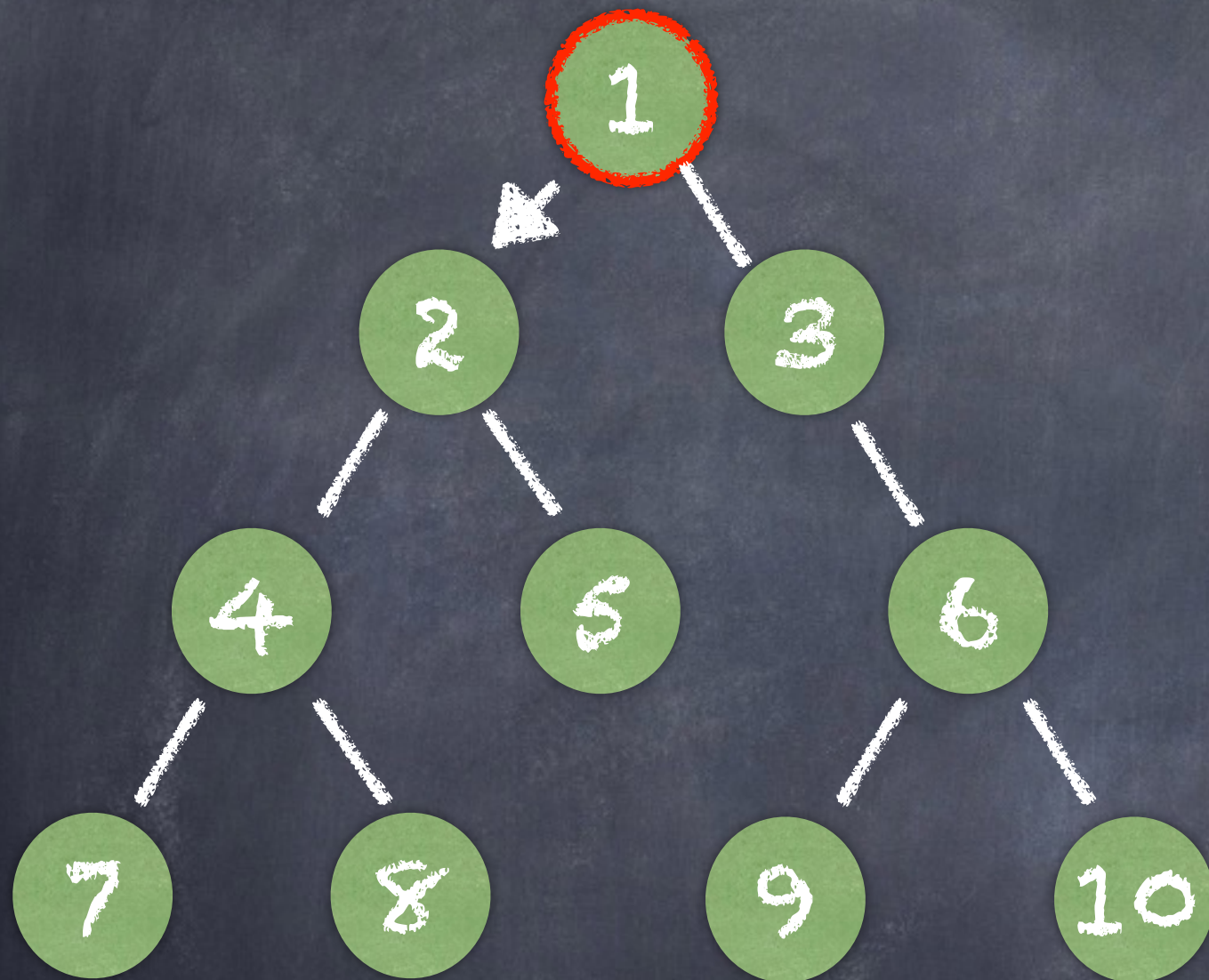
Pizzabestellung

Wiederholung
BFS



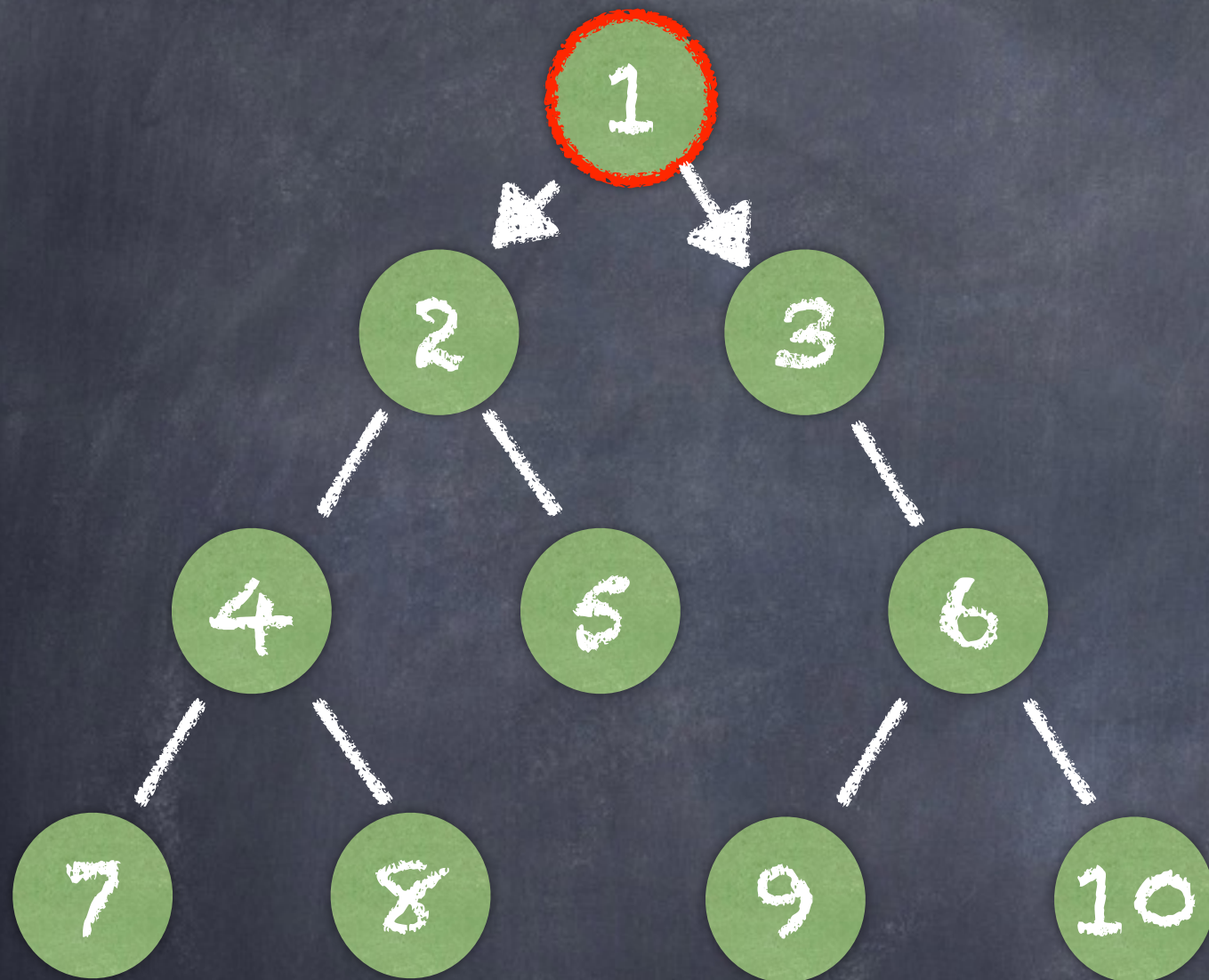
Pizzabestellung

Wiederholung
BFS



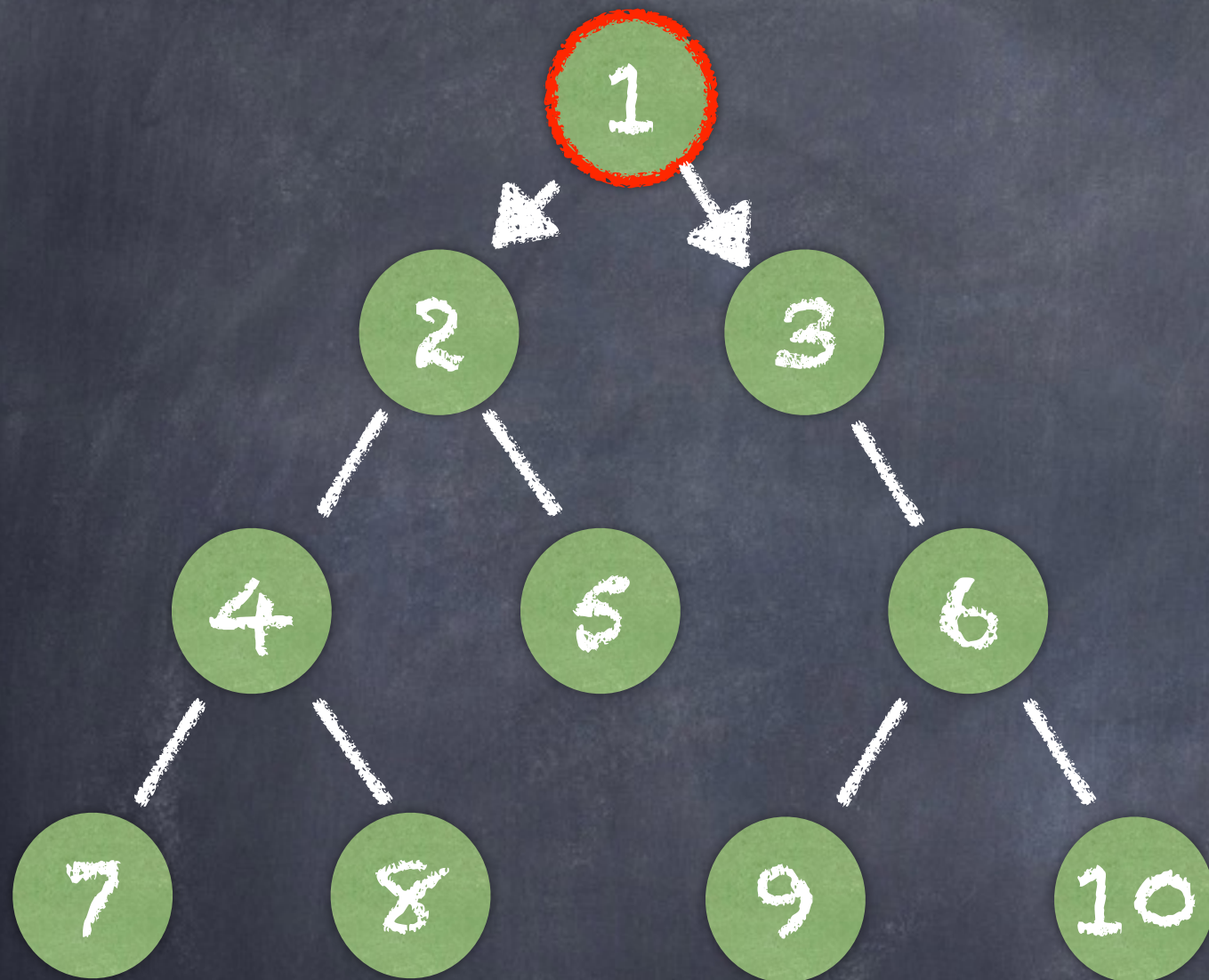
Pizzabestellung

Wiederholung
BFS



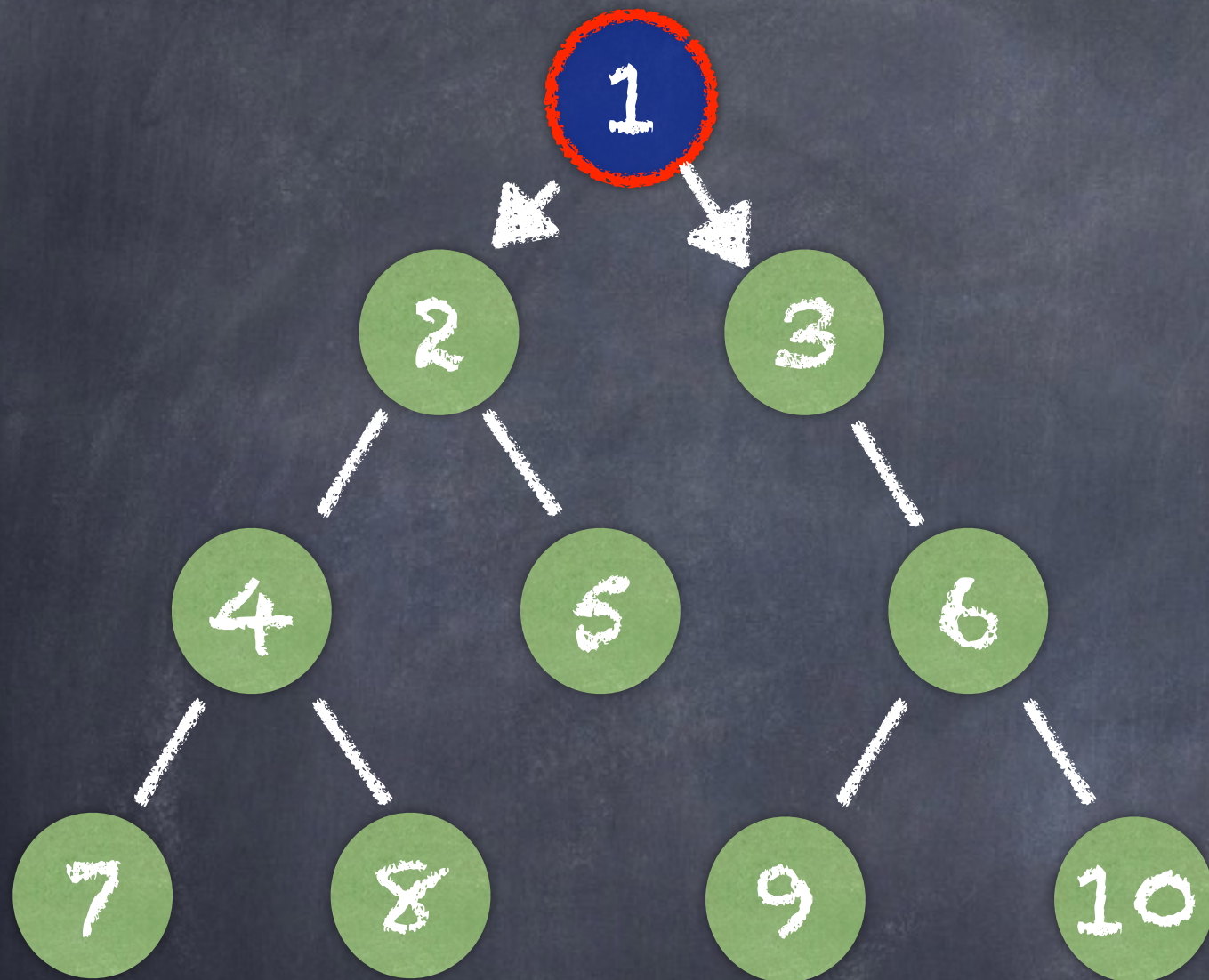
Pizzabestellung

Wiederholung
BFS



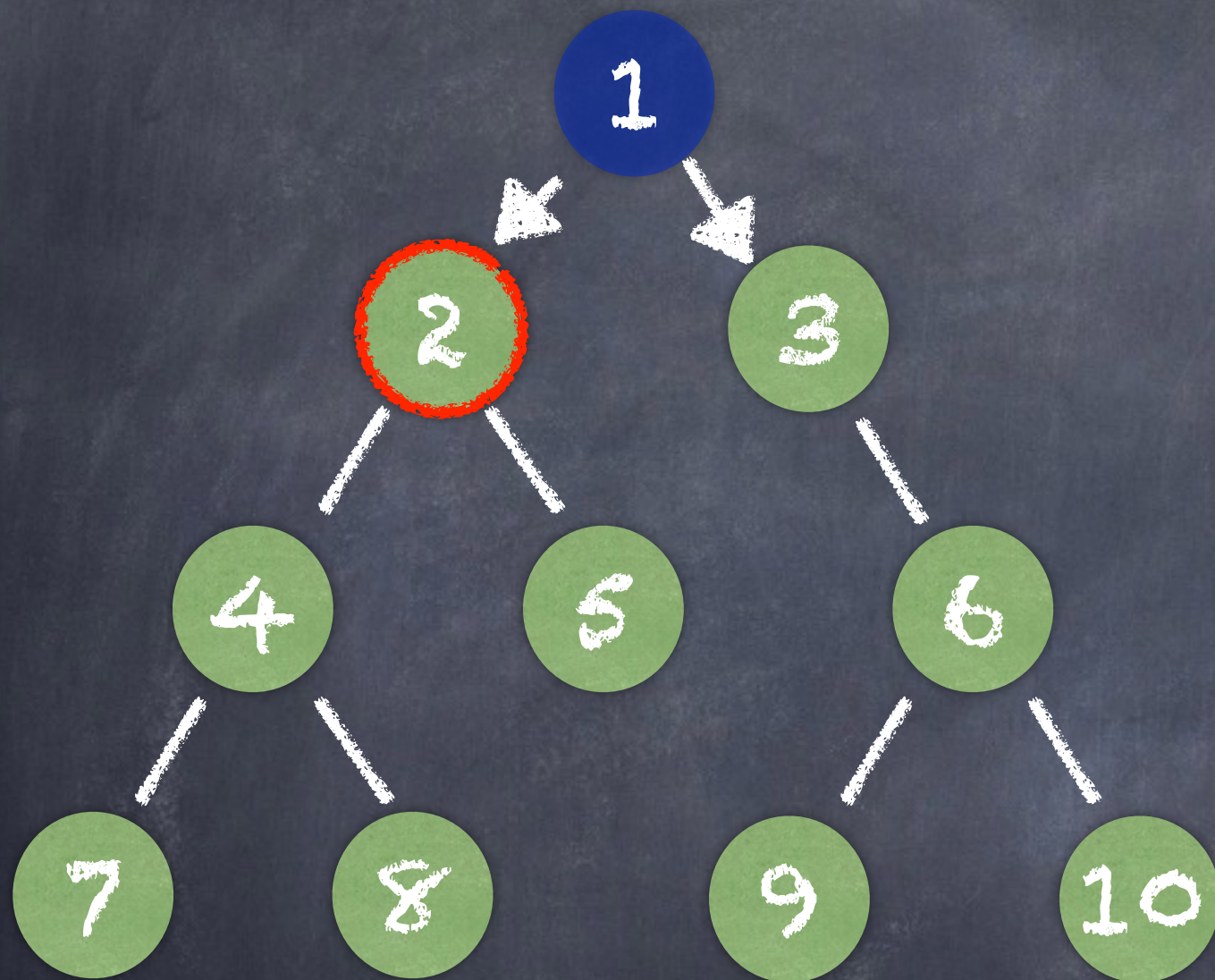
Pizzabestellung

Wiederholung
BFS



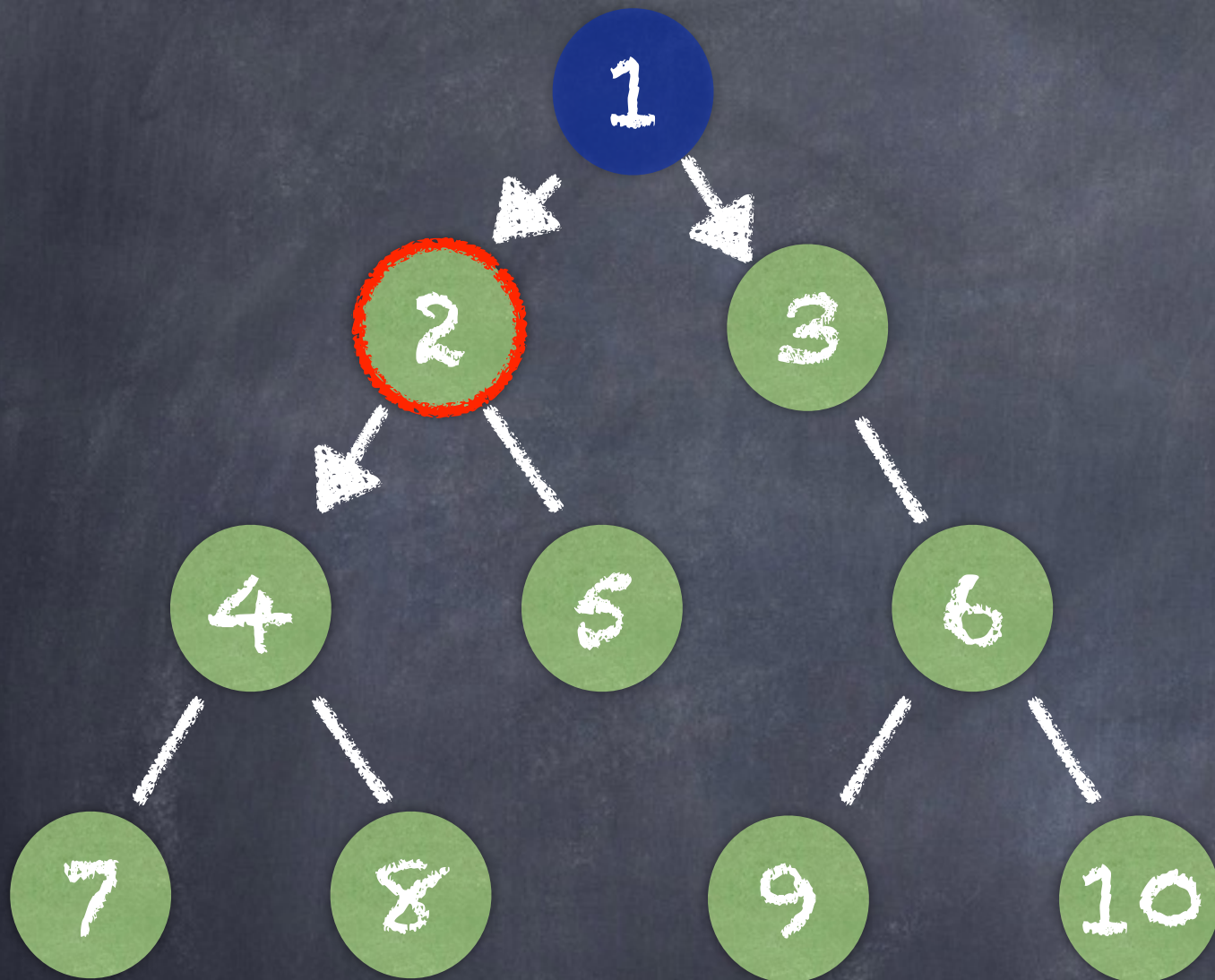
Pizzabestellung

Wiederholung
BFS



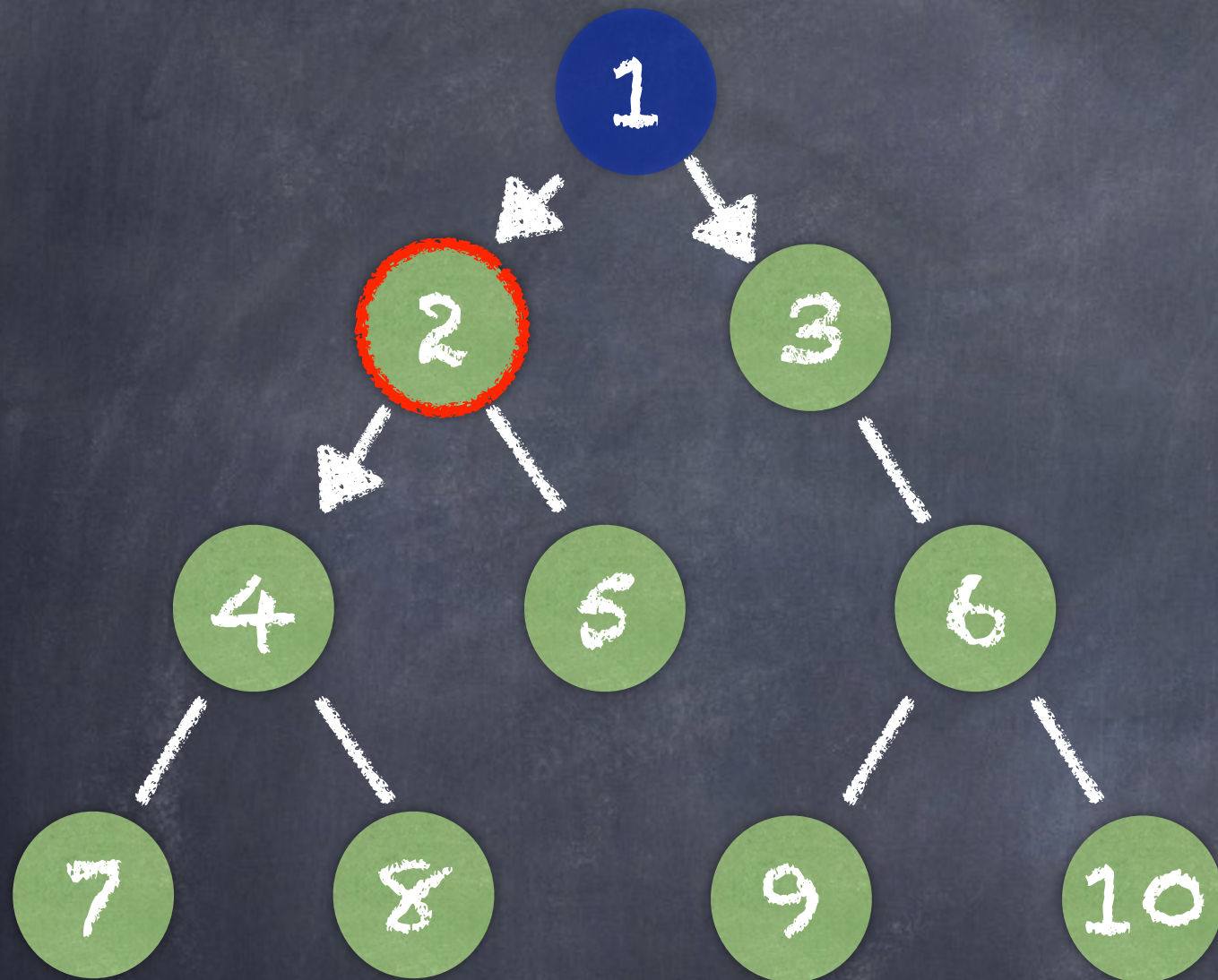
Pizzabestellung

Wiederholung
BFS



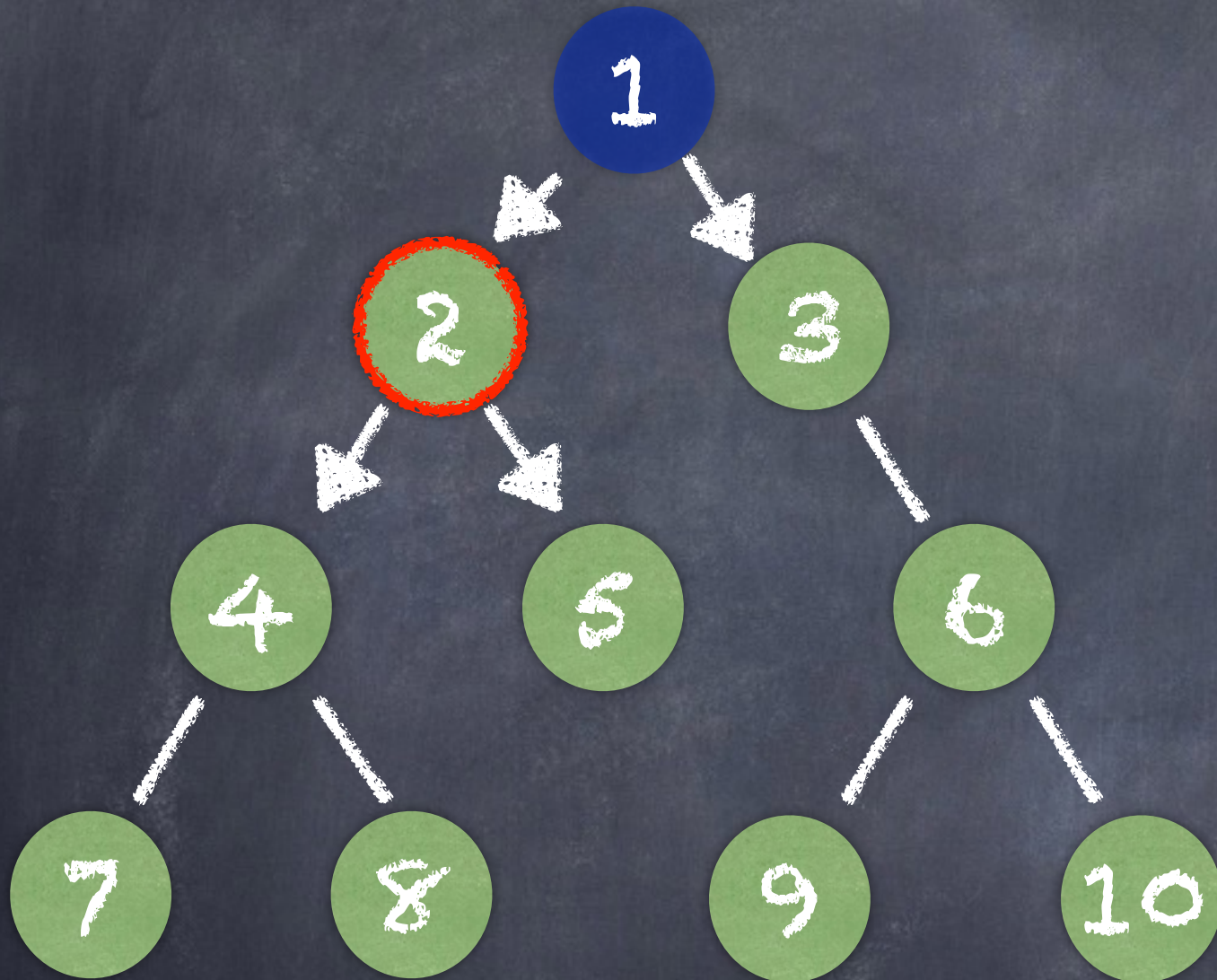
Pizzabestellung

Wiederholung
BFS



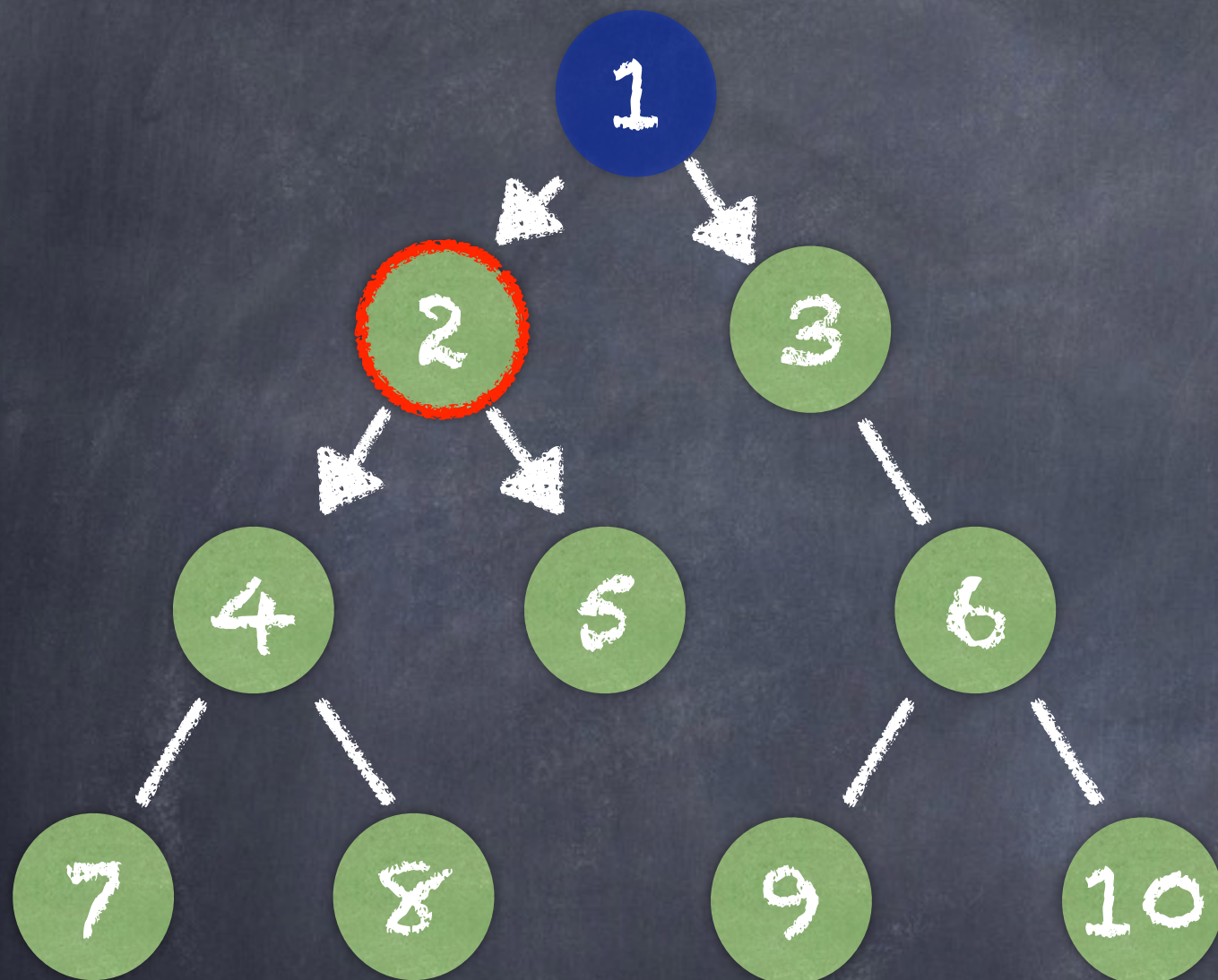
Pizzabestellung

Wiederholung
BFS



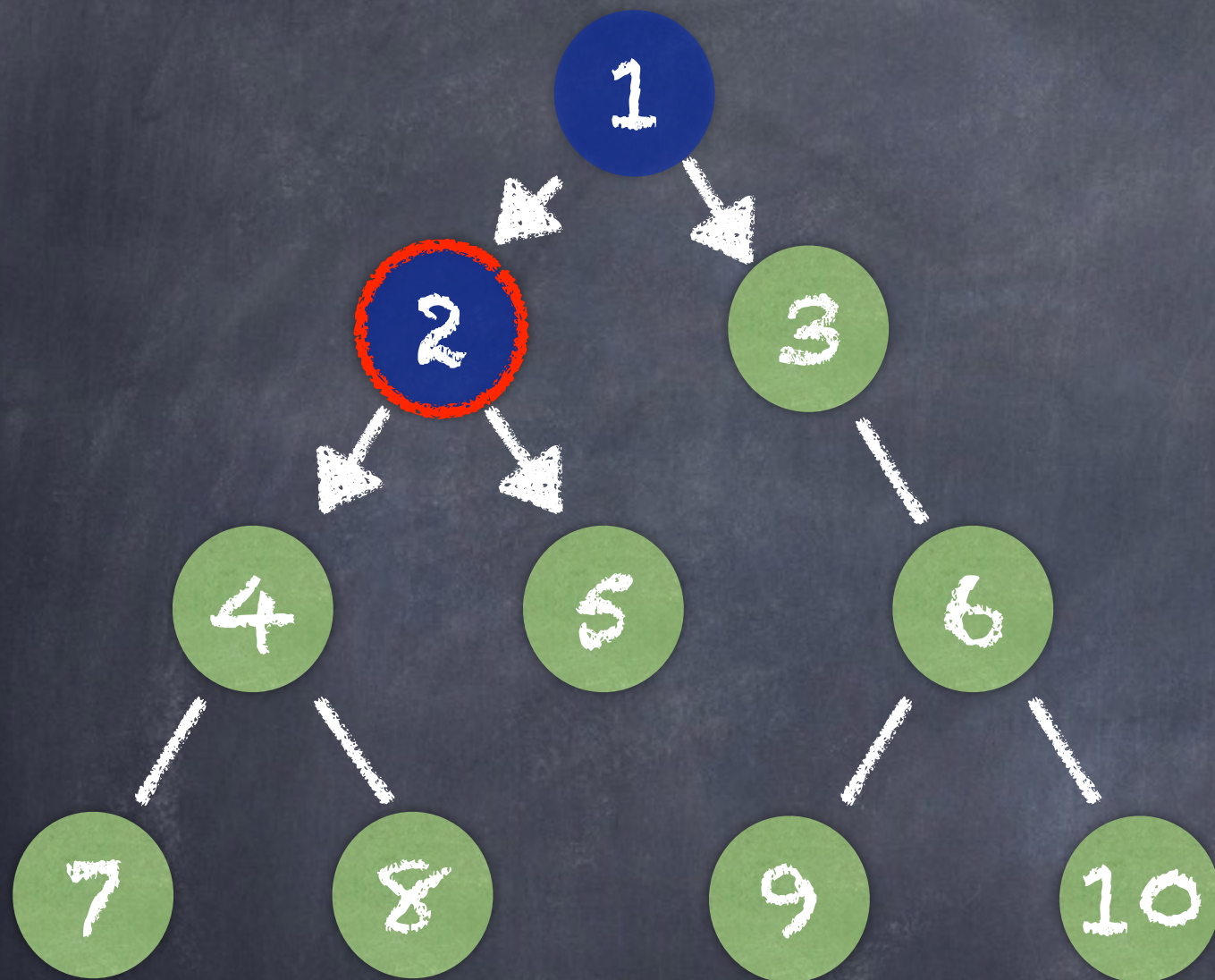
Pizzabestellung

Wiederholung
BFS



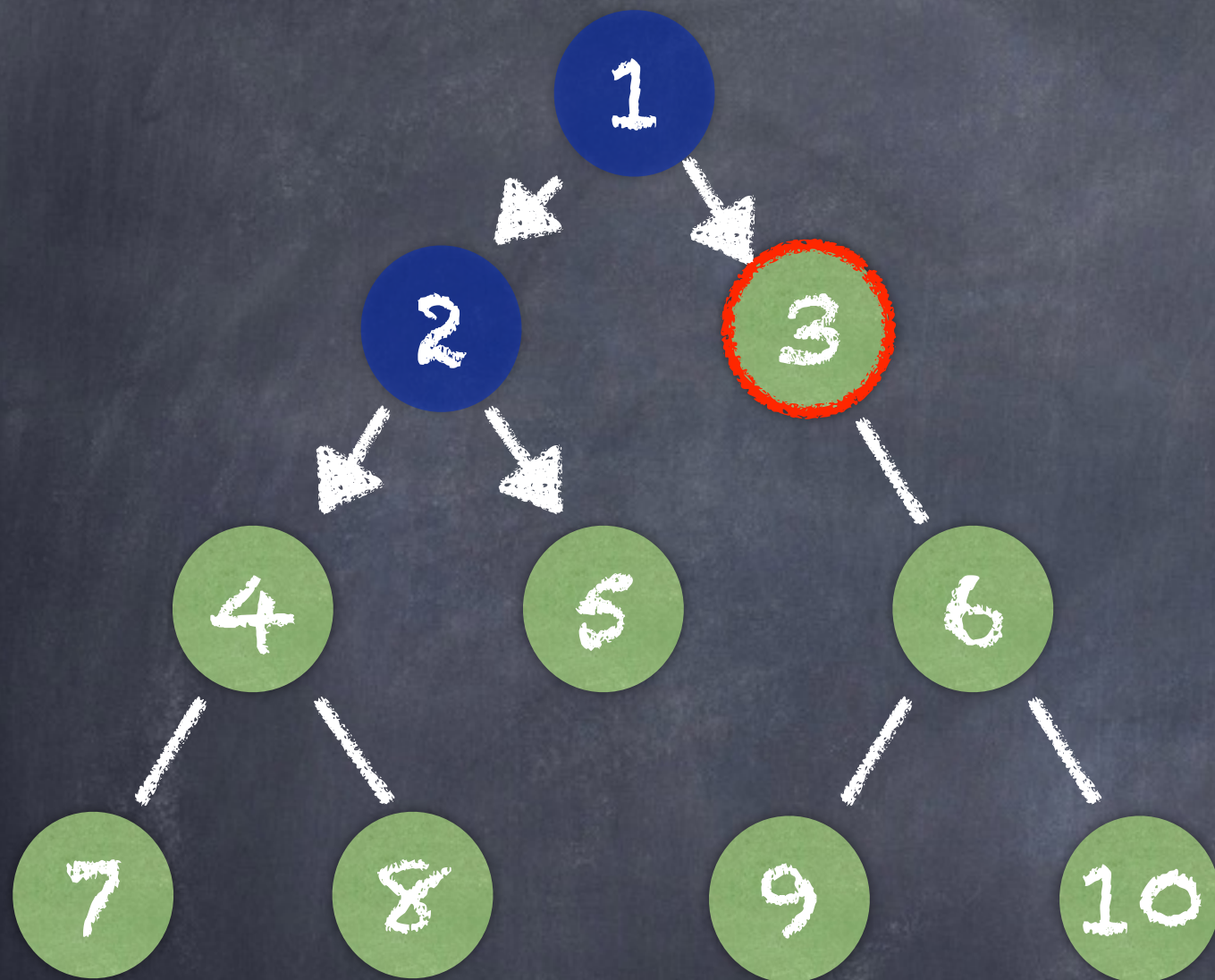
Pizzabestellung

Wiederholung
BFS



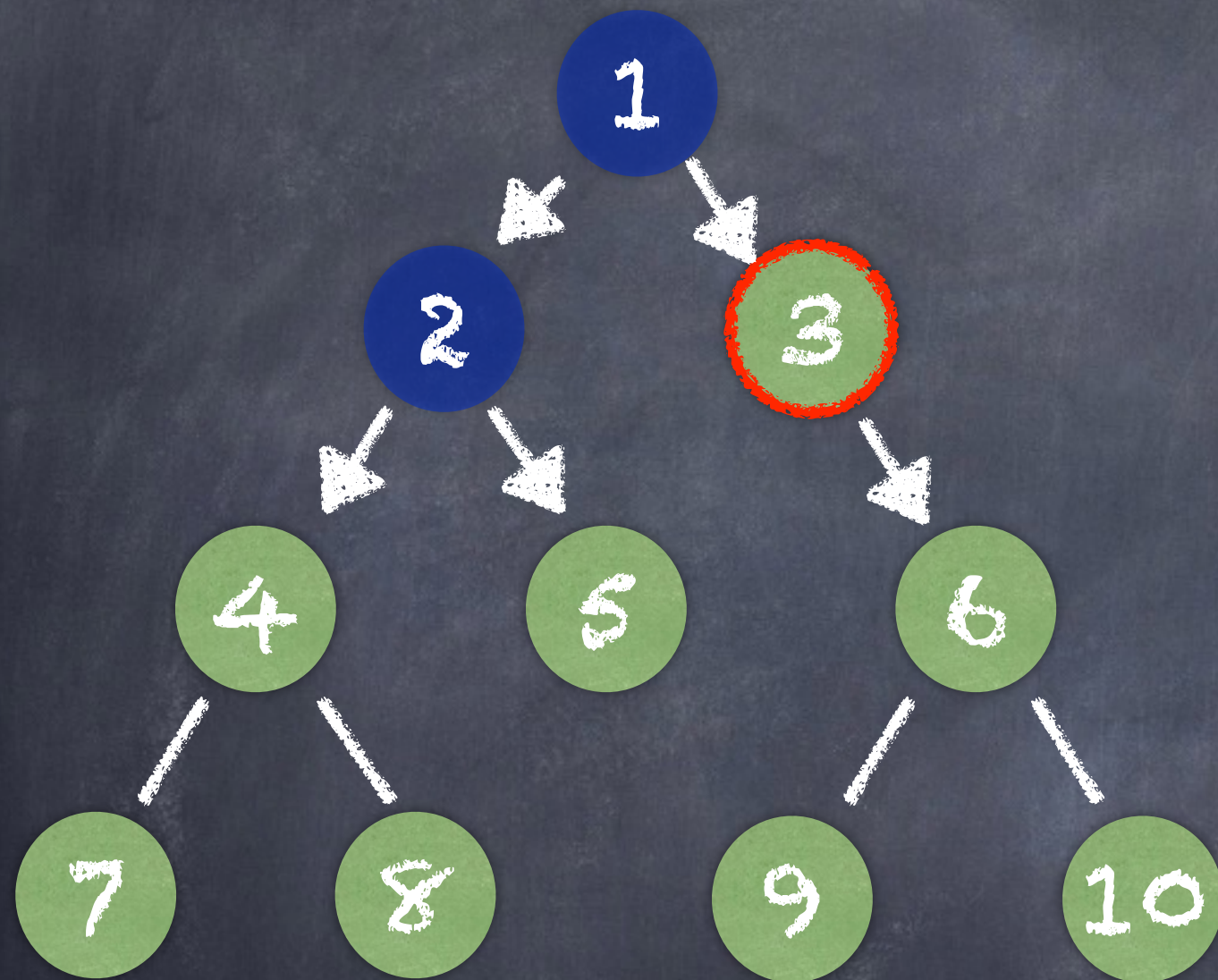
Pizzabestellung

Wiederholung
BFS



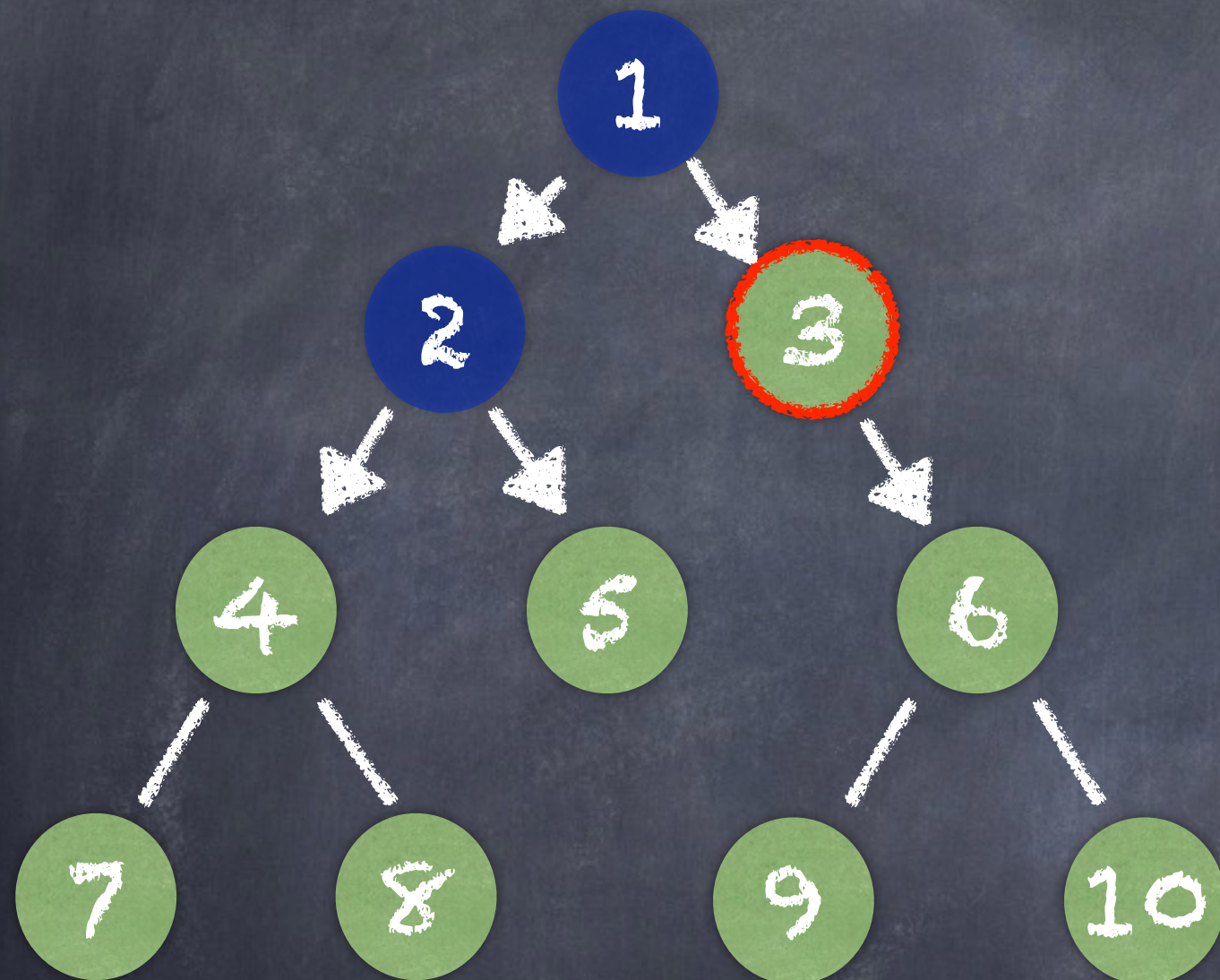
Pizzabestellung

Wiederholung
BFS



Pizzabestellung

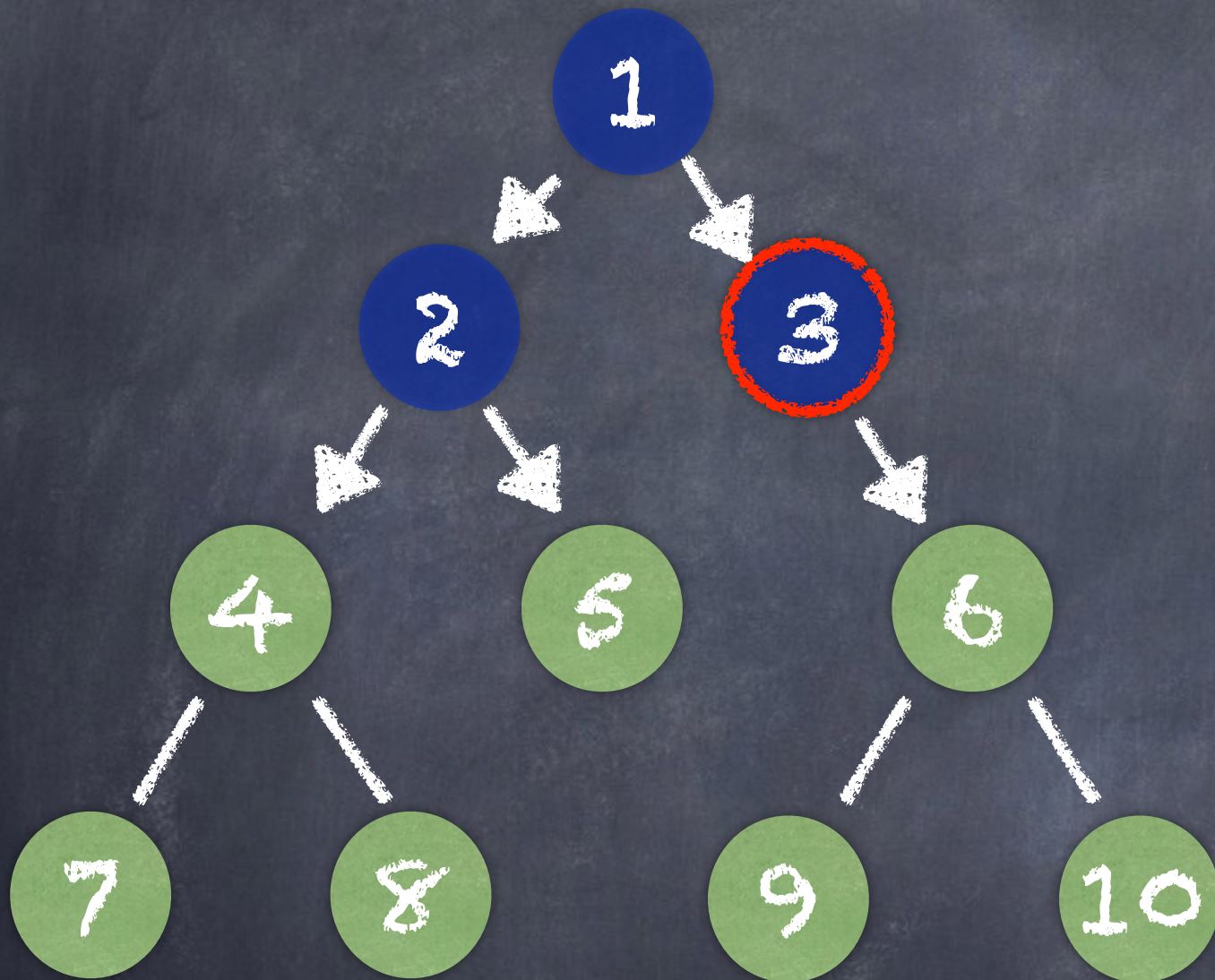
Wiederholung
BFS



Pizzabestellung

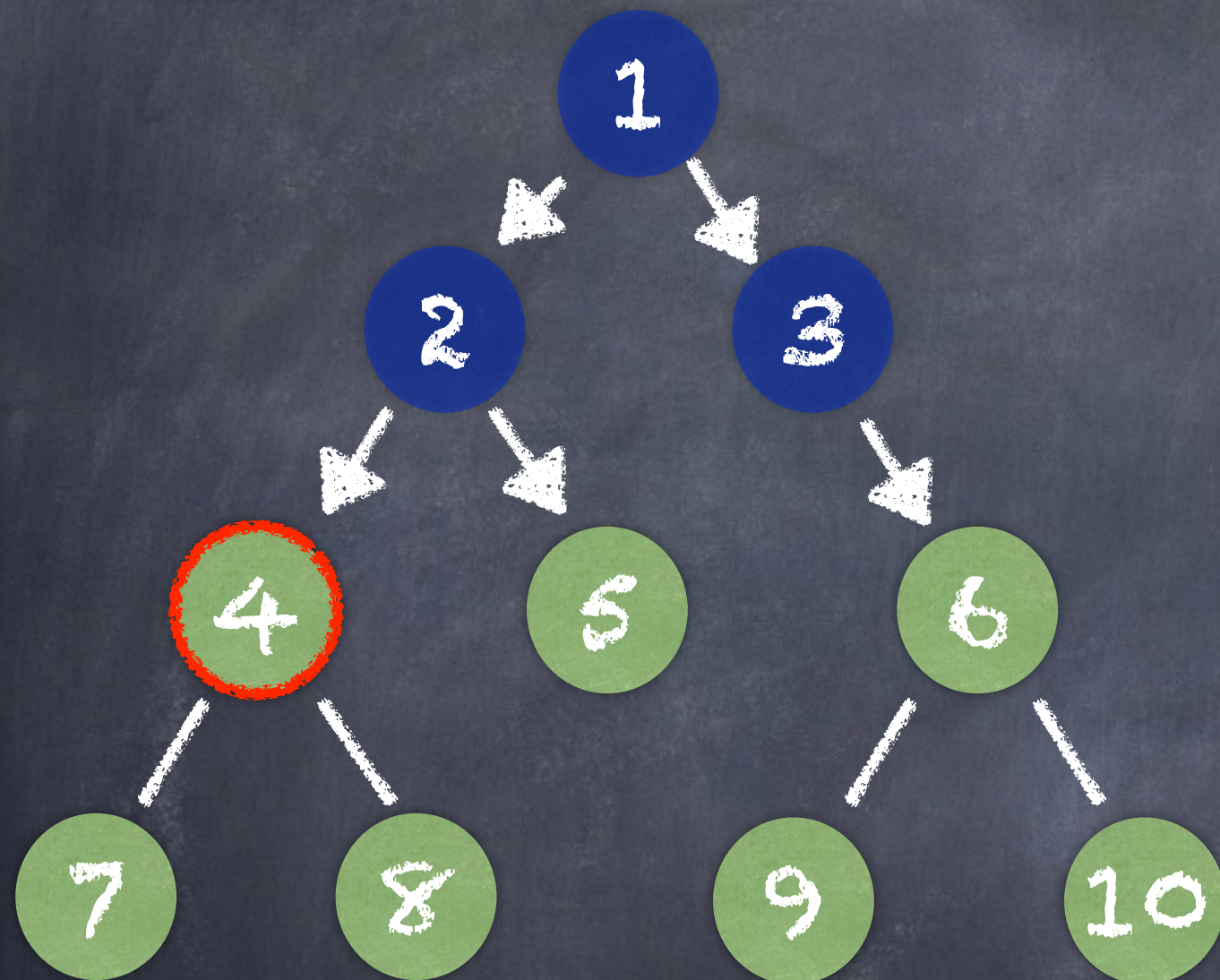
Wiederholung

BFS



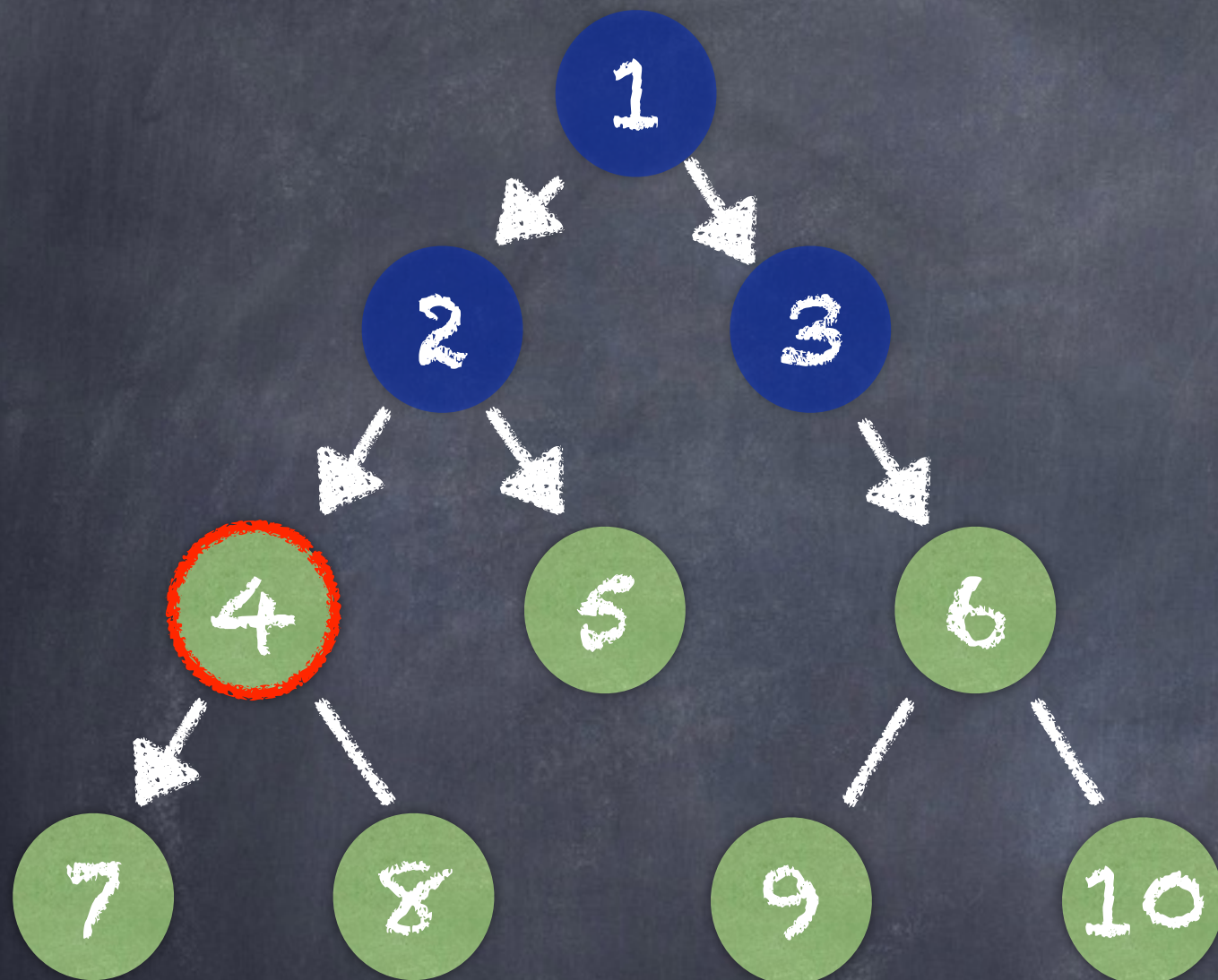
Pizzabestellung

Wiederholung
BFS



Pizzabestellung

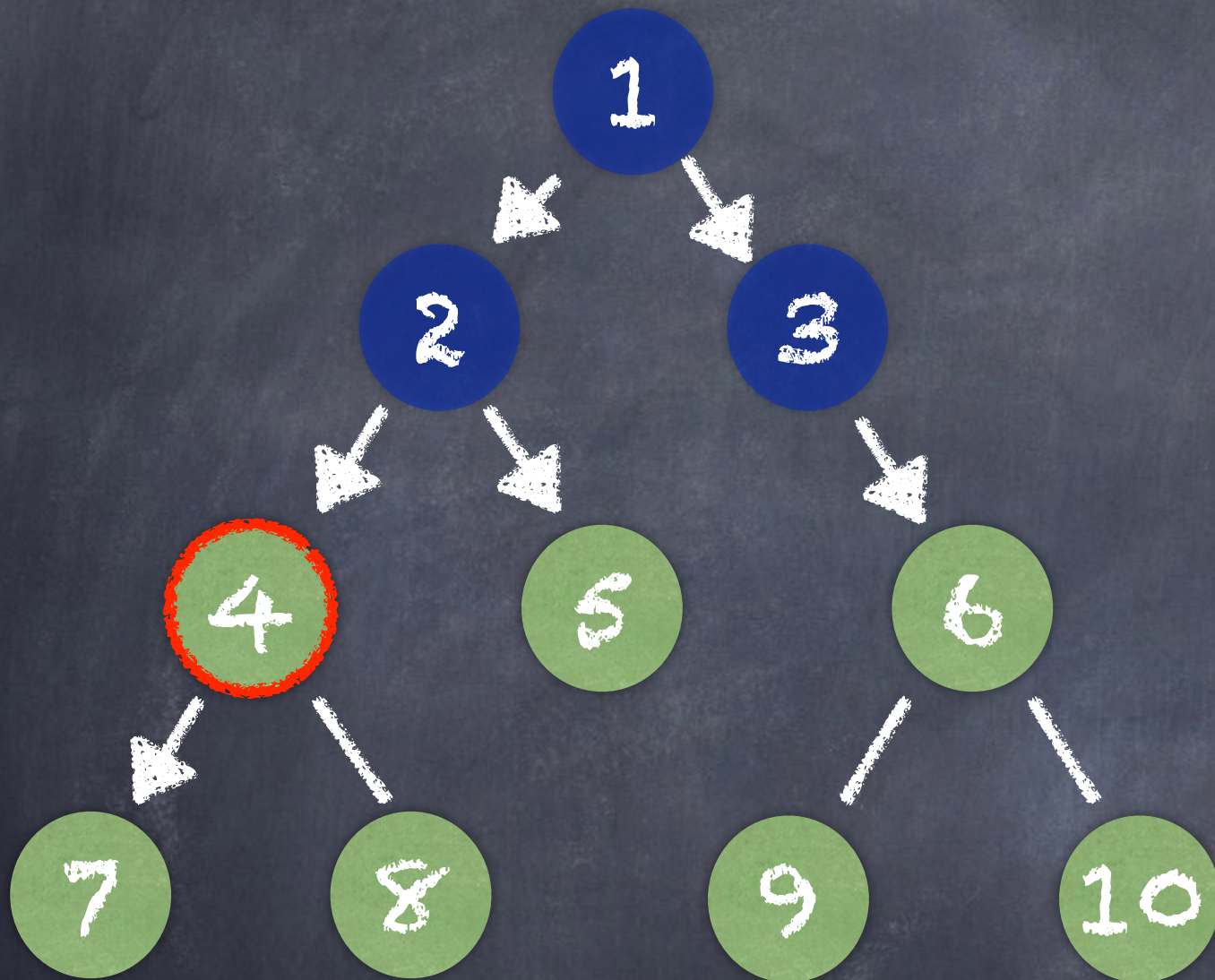
Wiederholung
BFS



Pizzabestellung

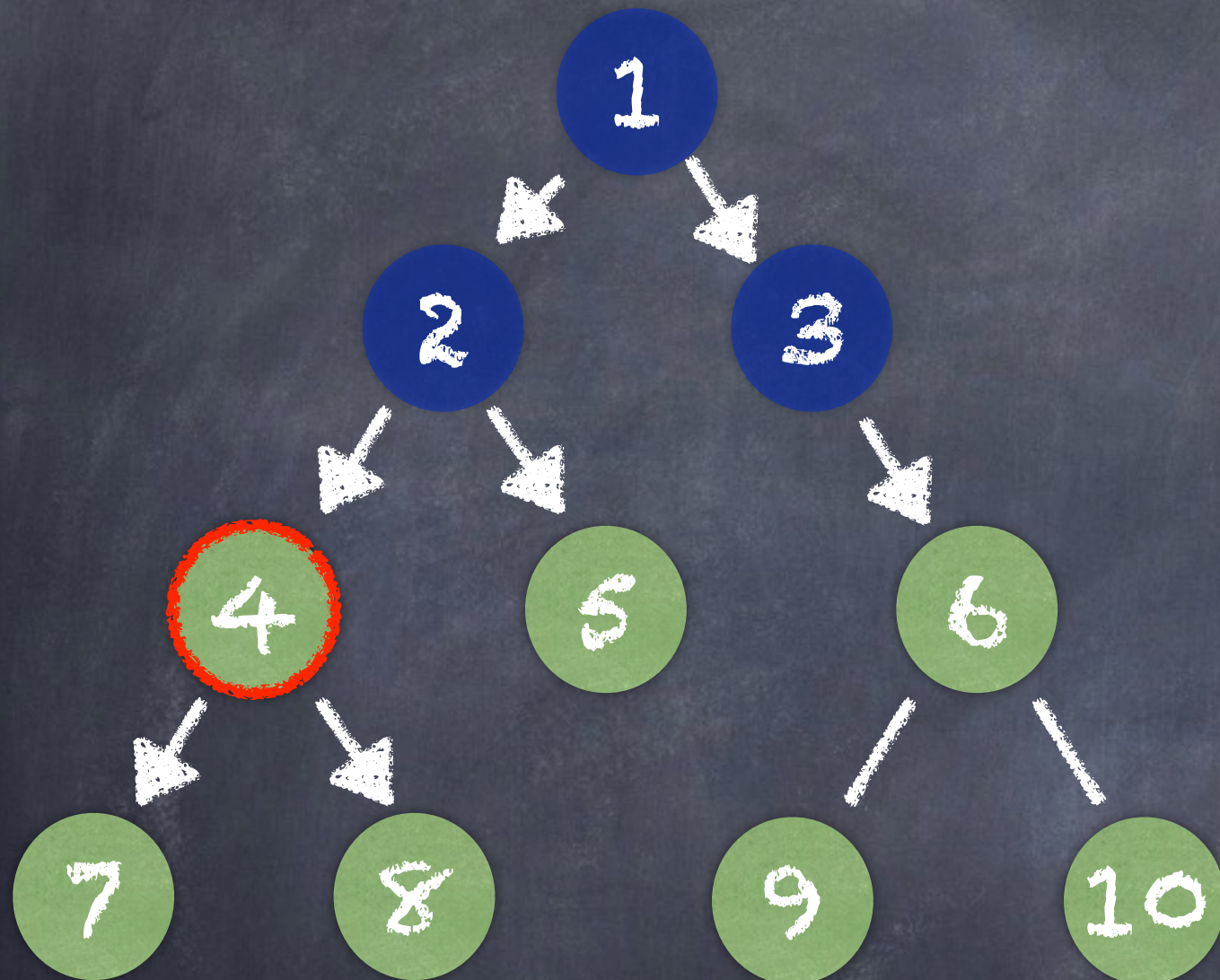
Wiederholung

BFS



Pizzabestellung

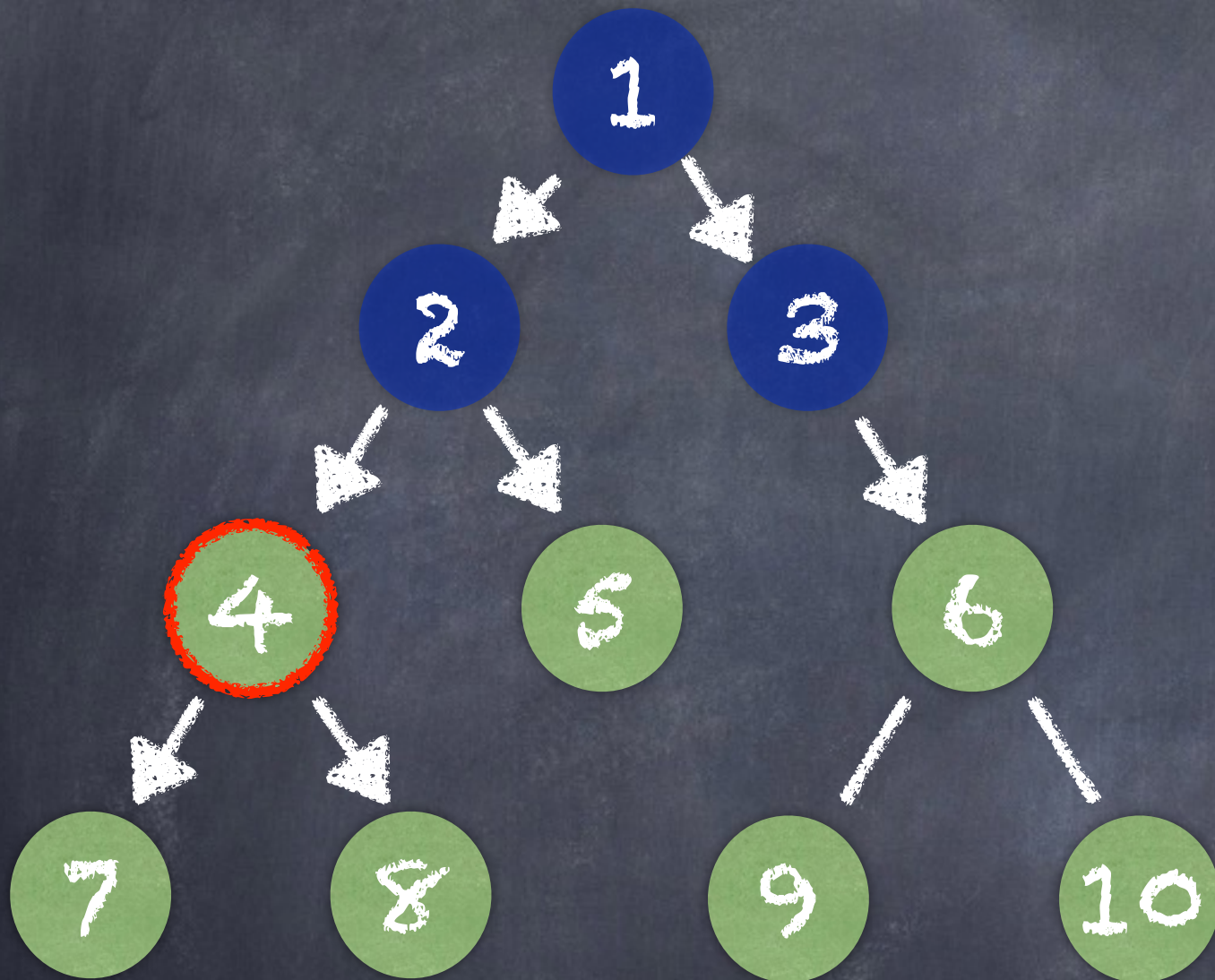
Wiederholung
BFS



Pizzabestellung

Wiederholung

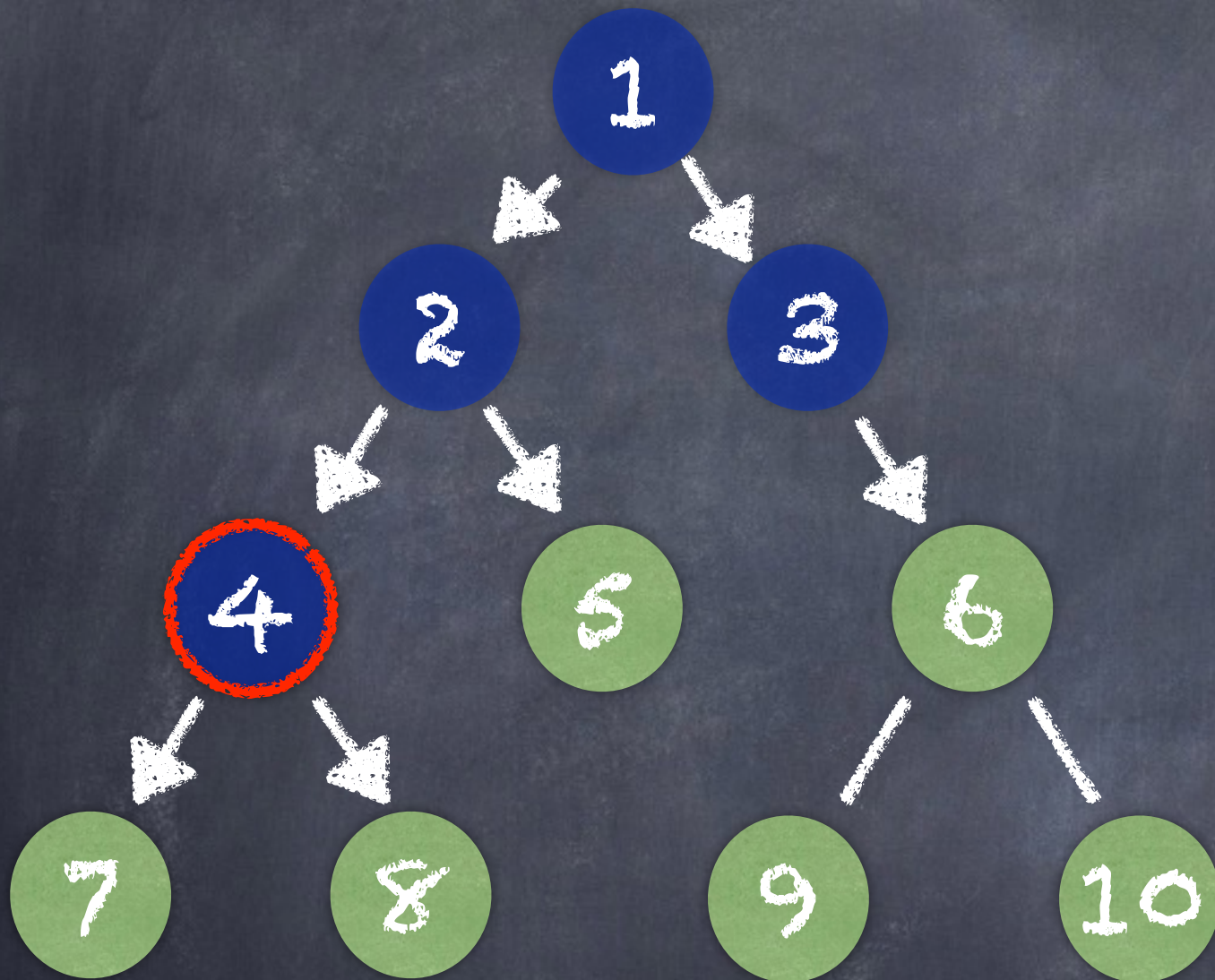
BFS



Pizzabestellung

Wiederholung

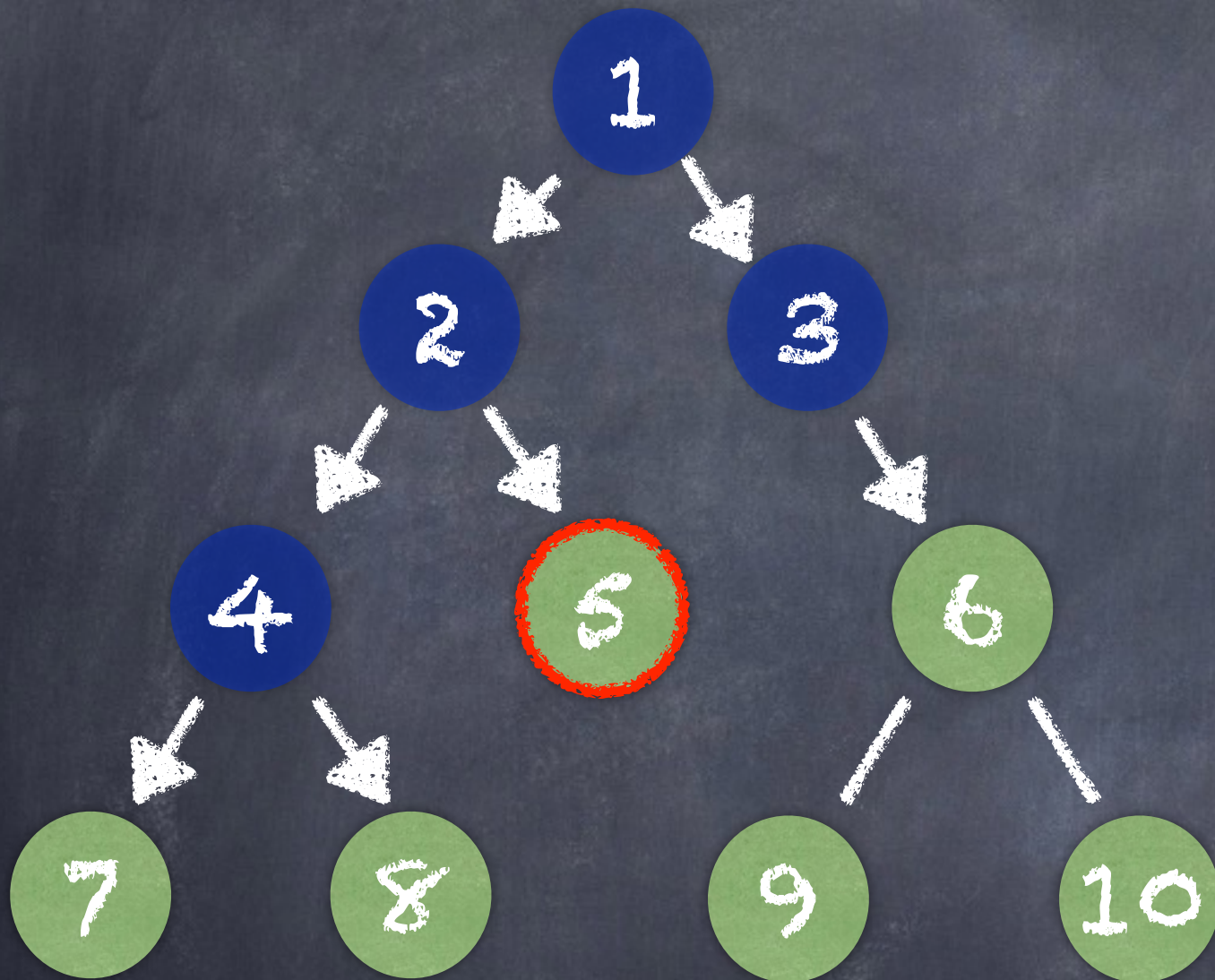
BFS



Pizzabestellung

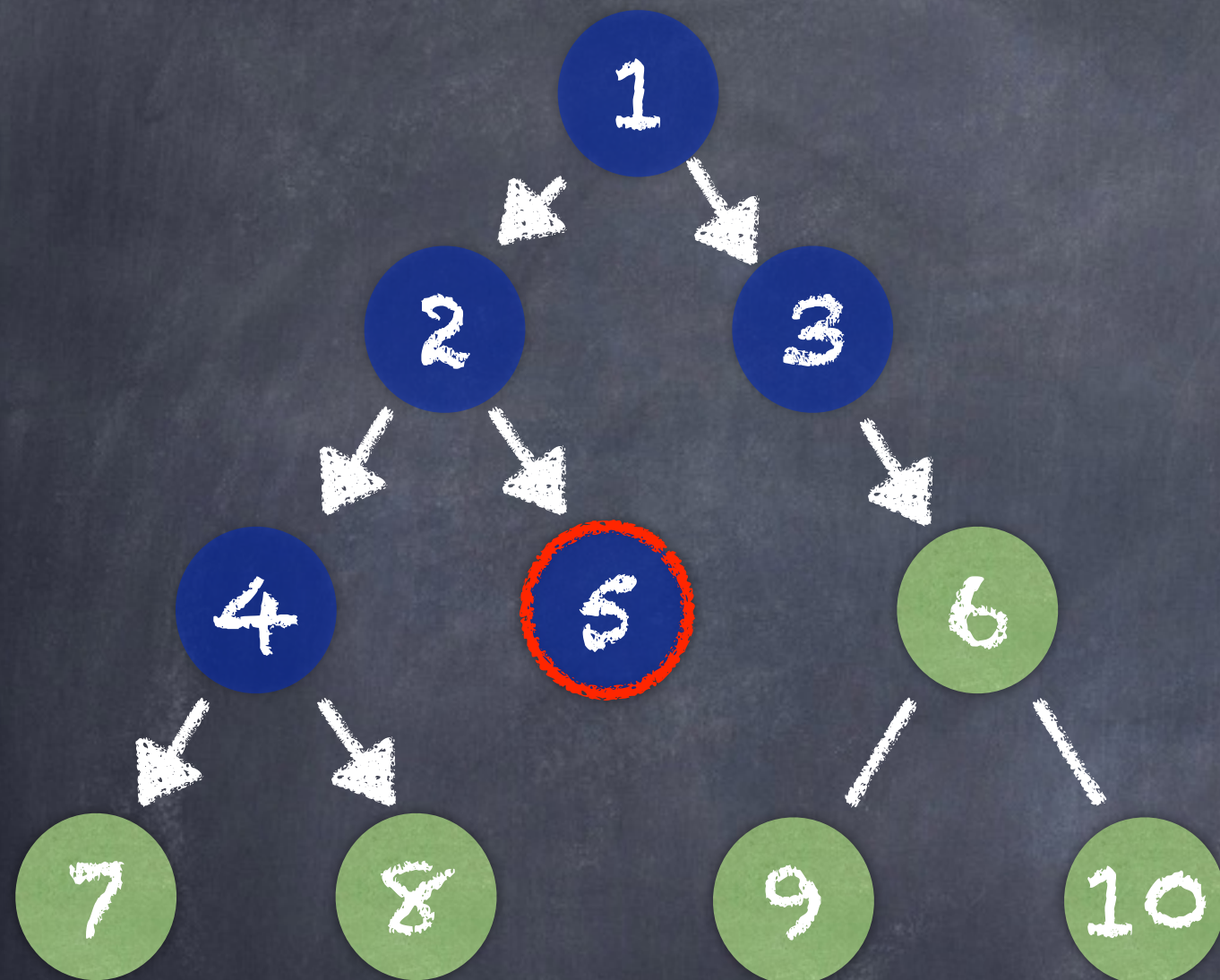
Wiederholung

BFS



Pizzabestellung

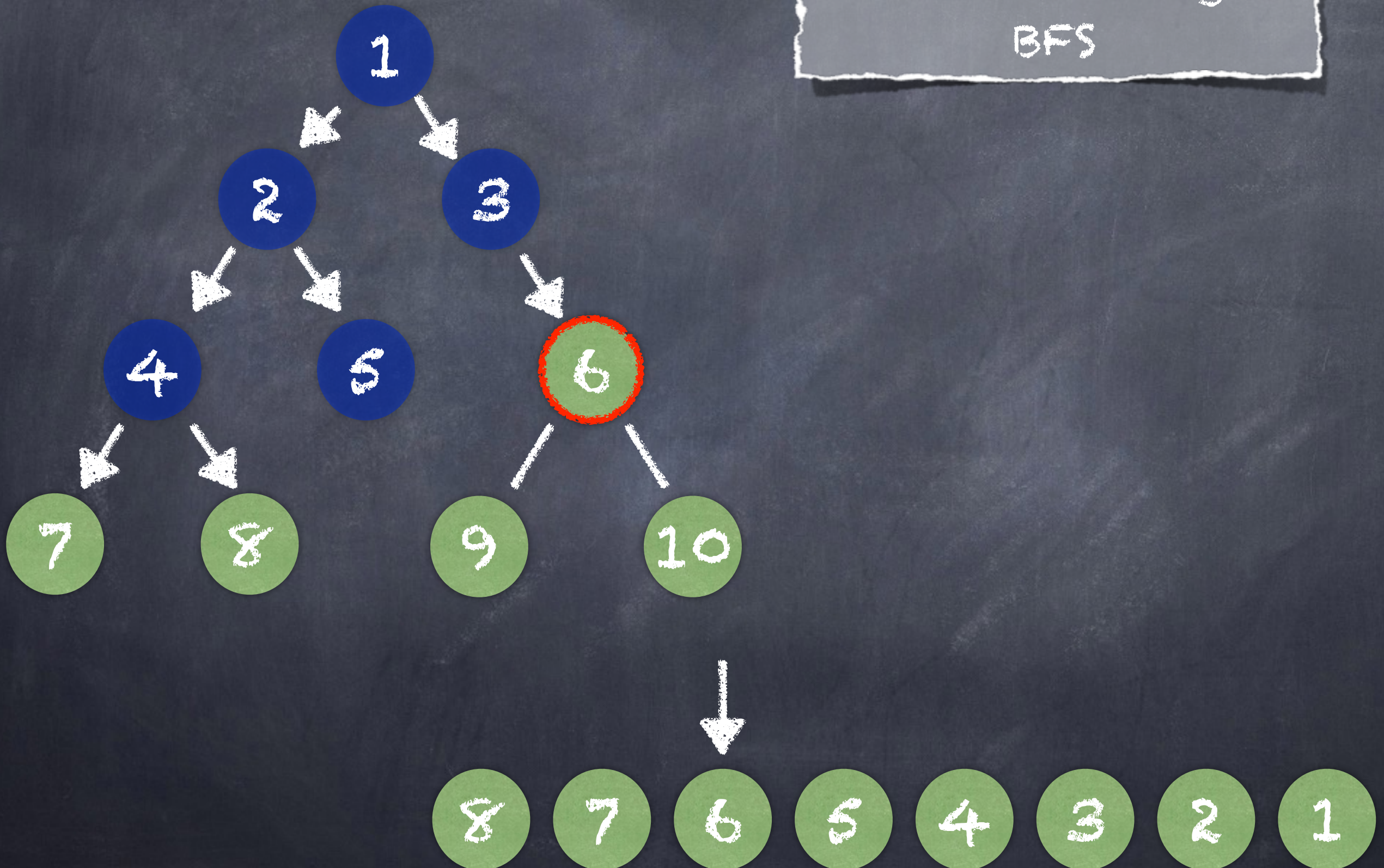
Wiederholung
BFS



Pizzabestellung

Wiederholung

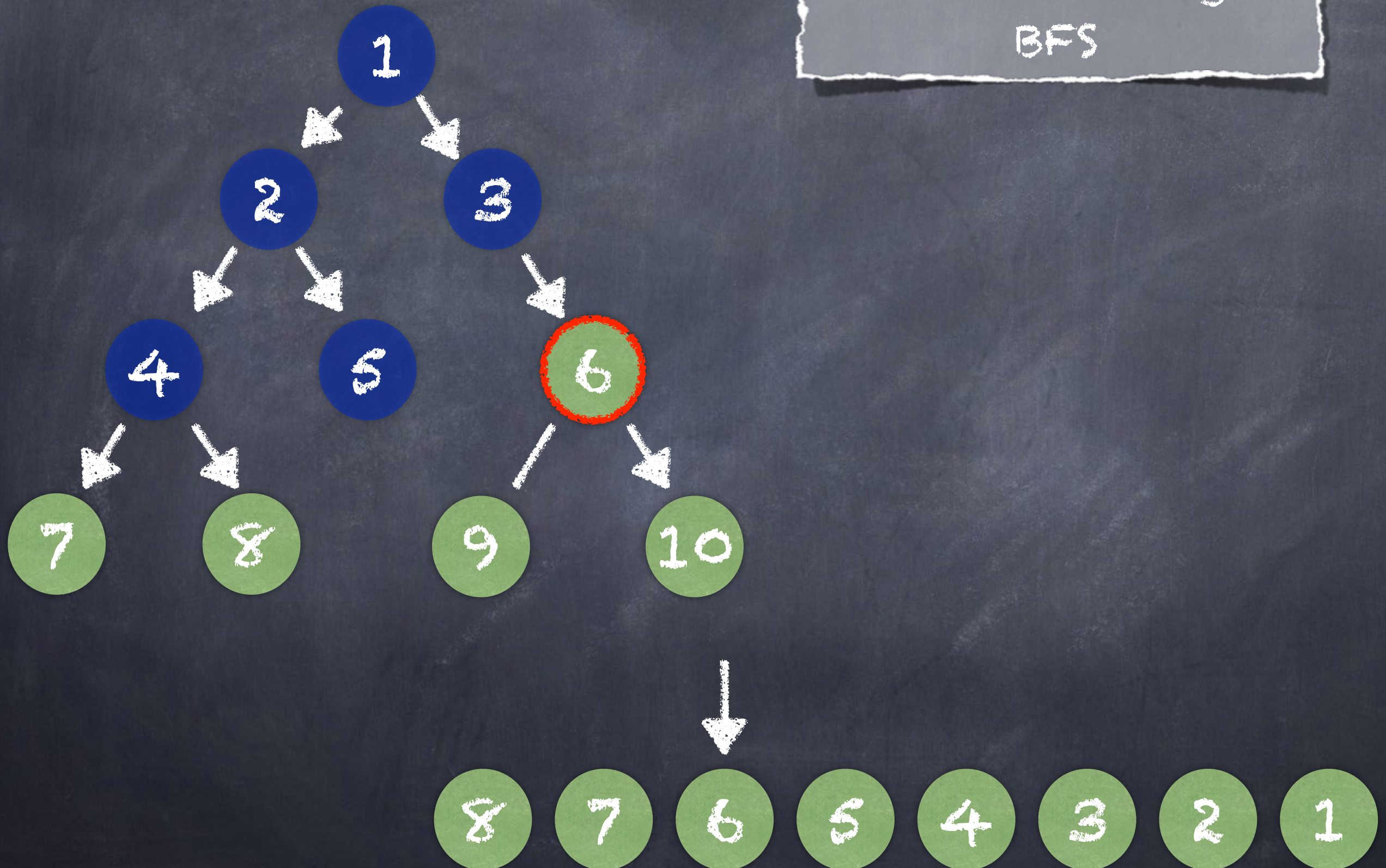
BFS



Pizzabestellung

Wiederholung

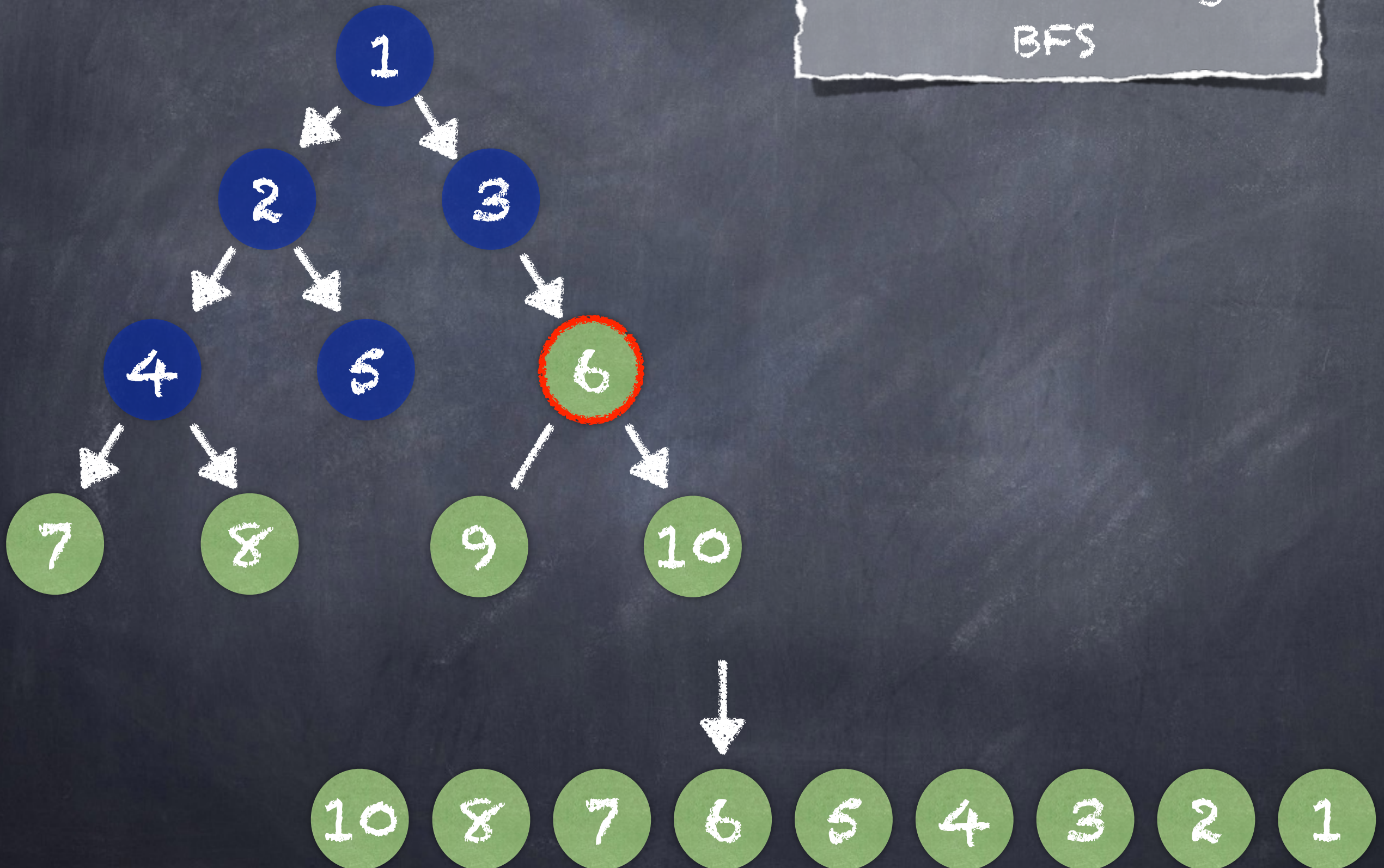
BFS



Pizzabestellung

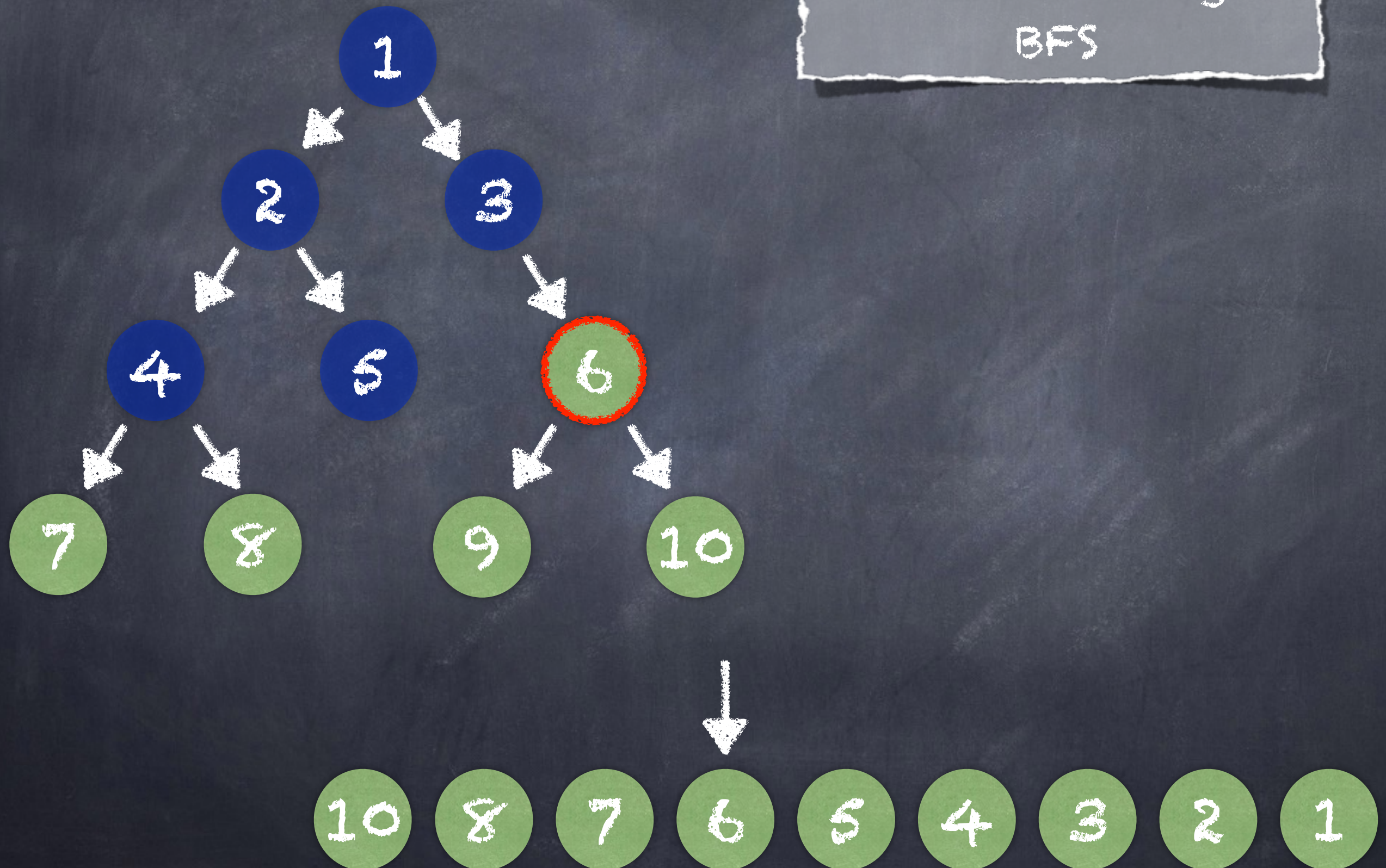
Wiederholung

BFS



Pizzabestellung

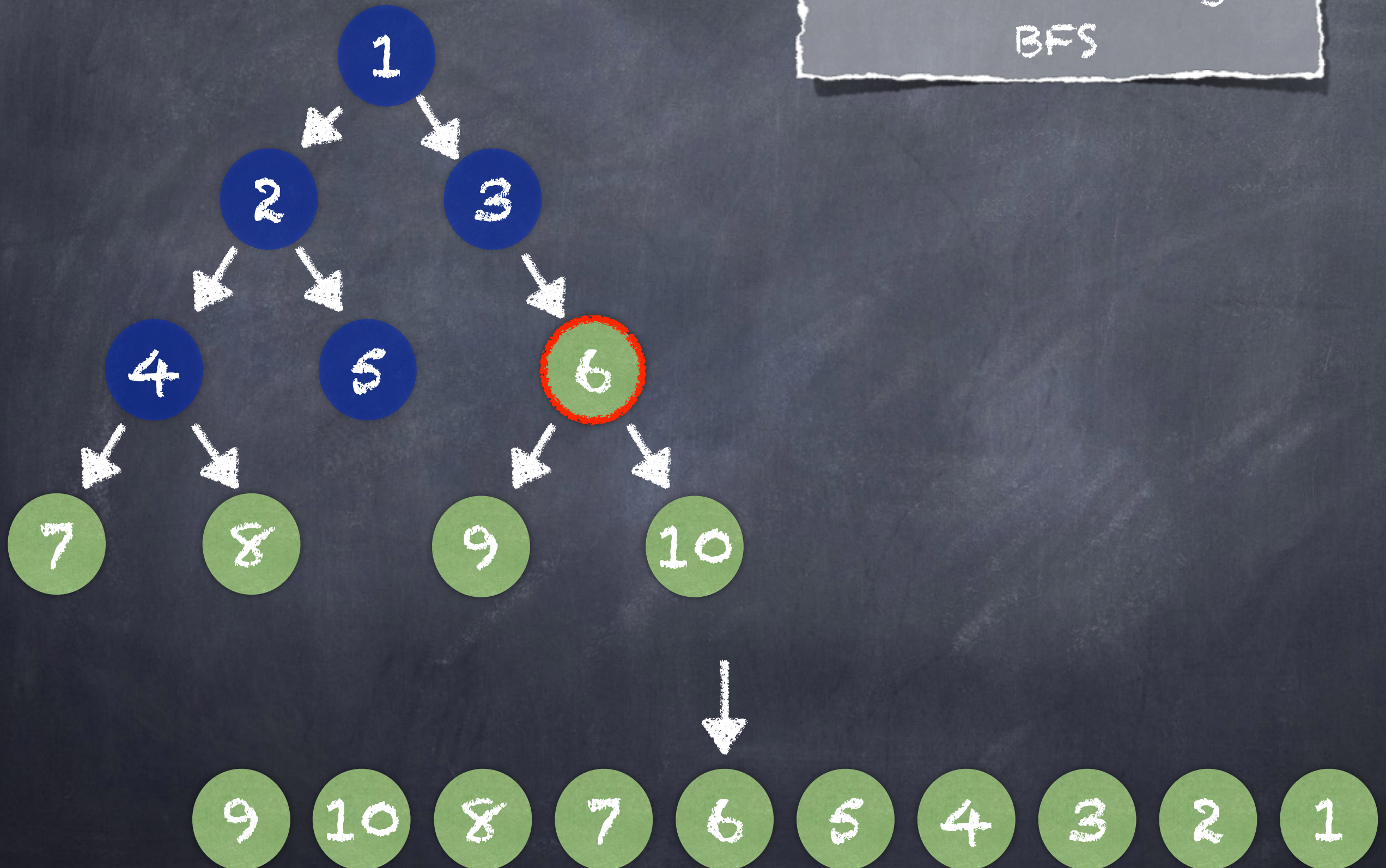
Wiederholung
BFS



Pizzabestellung

Wiederholung

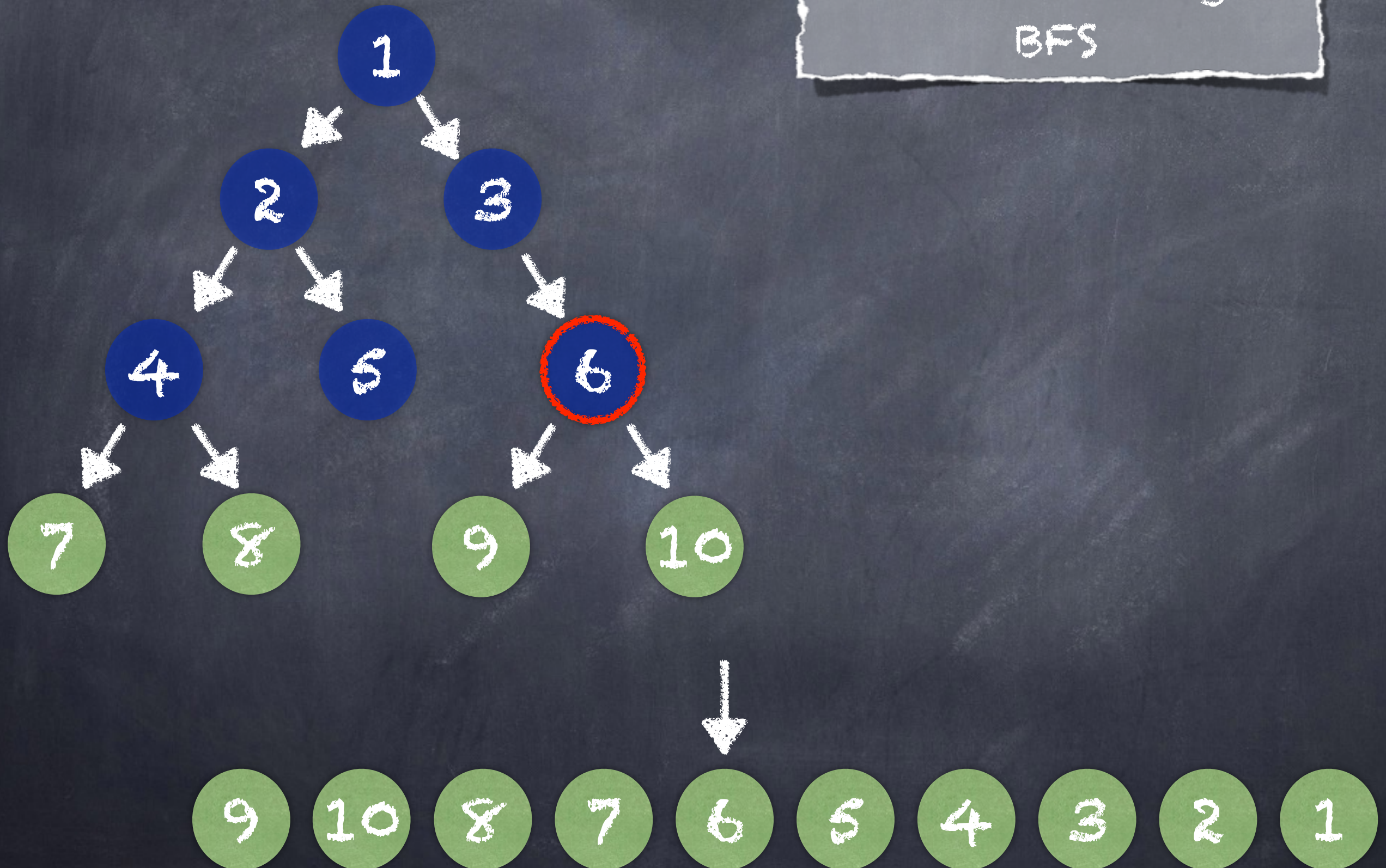
BFS



Pizzabestellung

Wiederholung

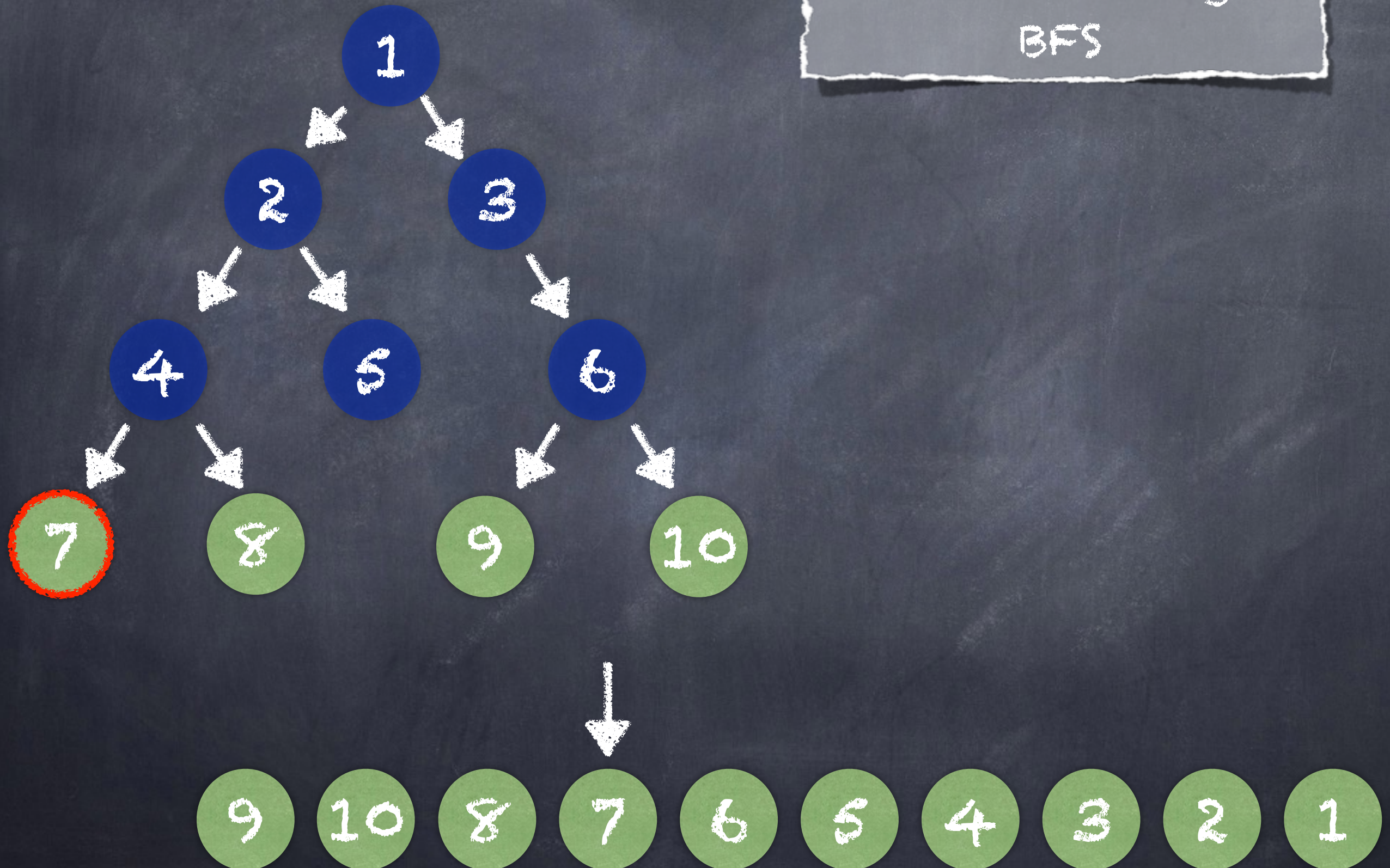
BFS



Pizzabestellung

Wiederholung

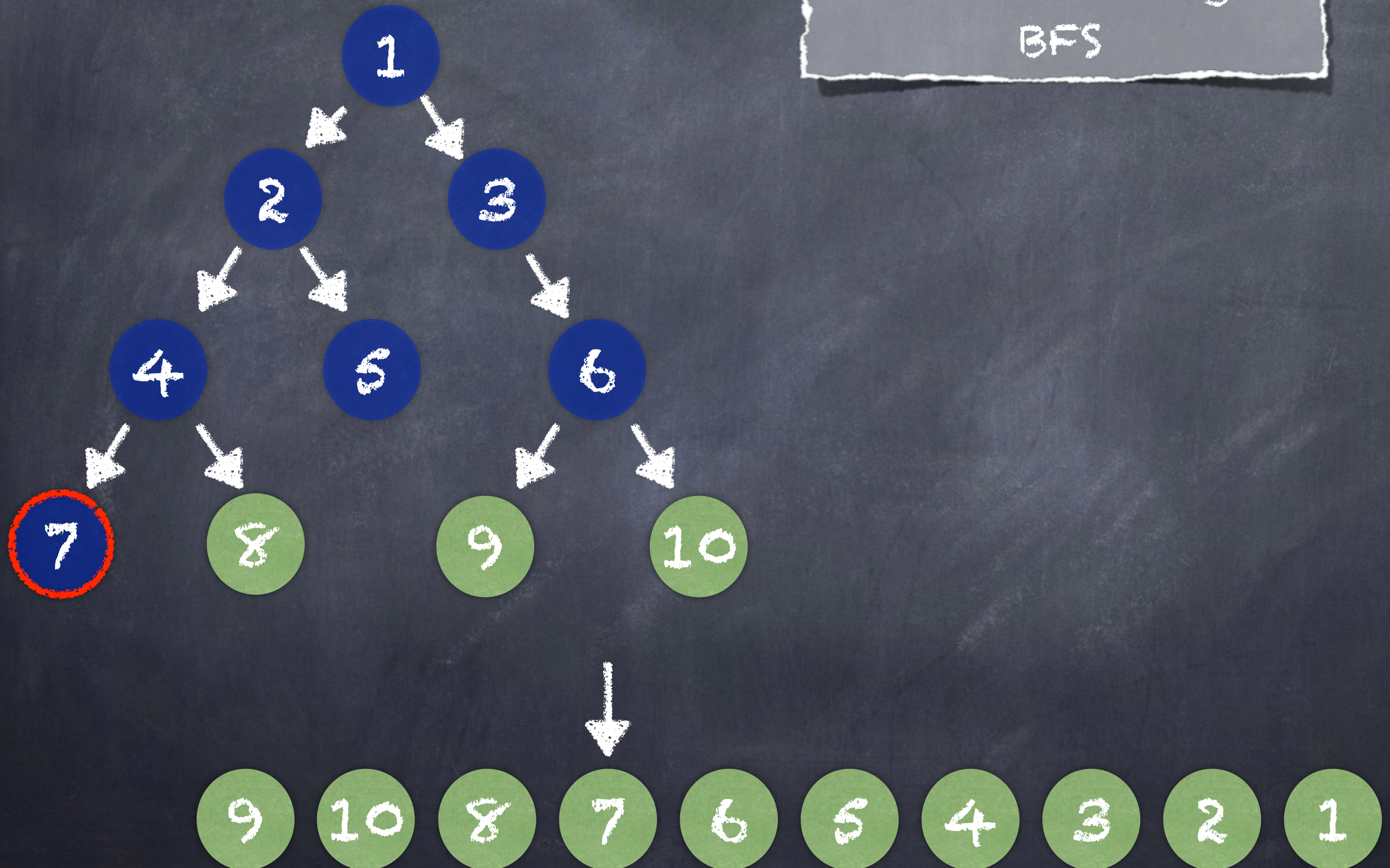
BFS



Pizzabestellung

Wiederholung

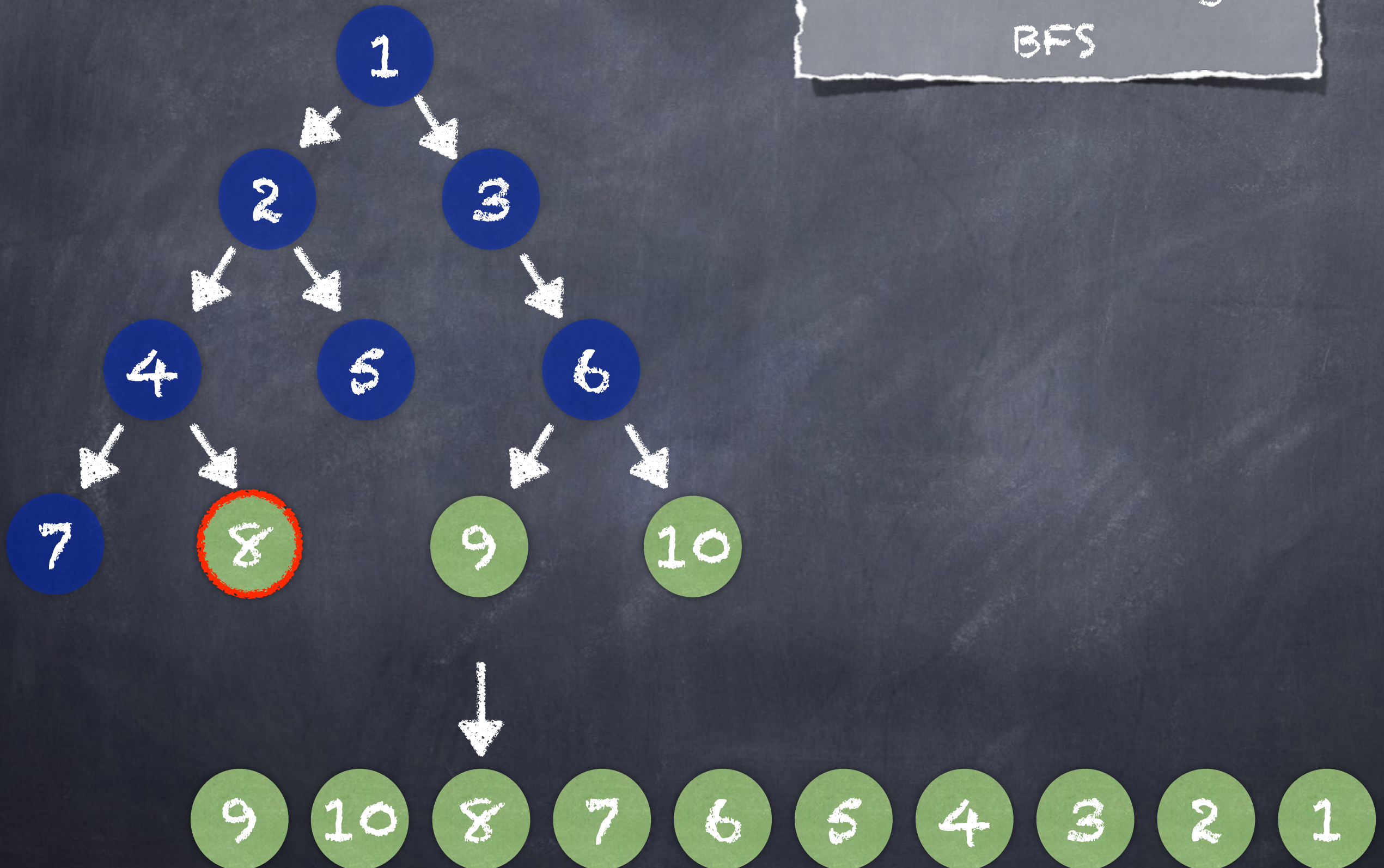
BFS



Pizzabestellung

Wiederholung

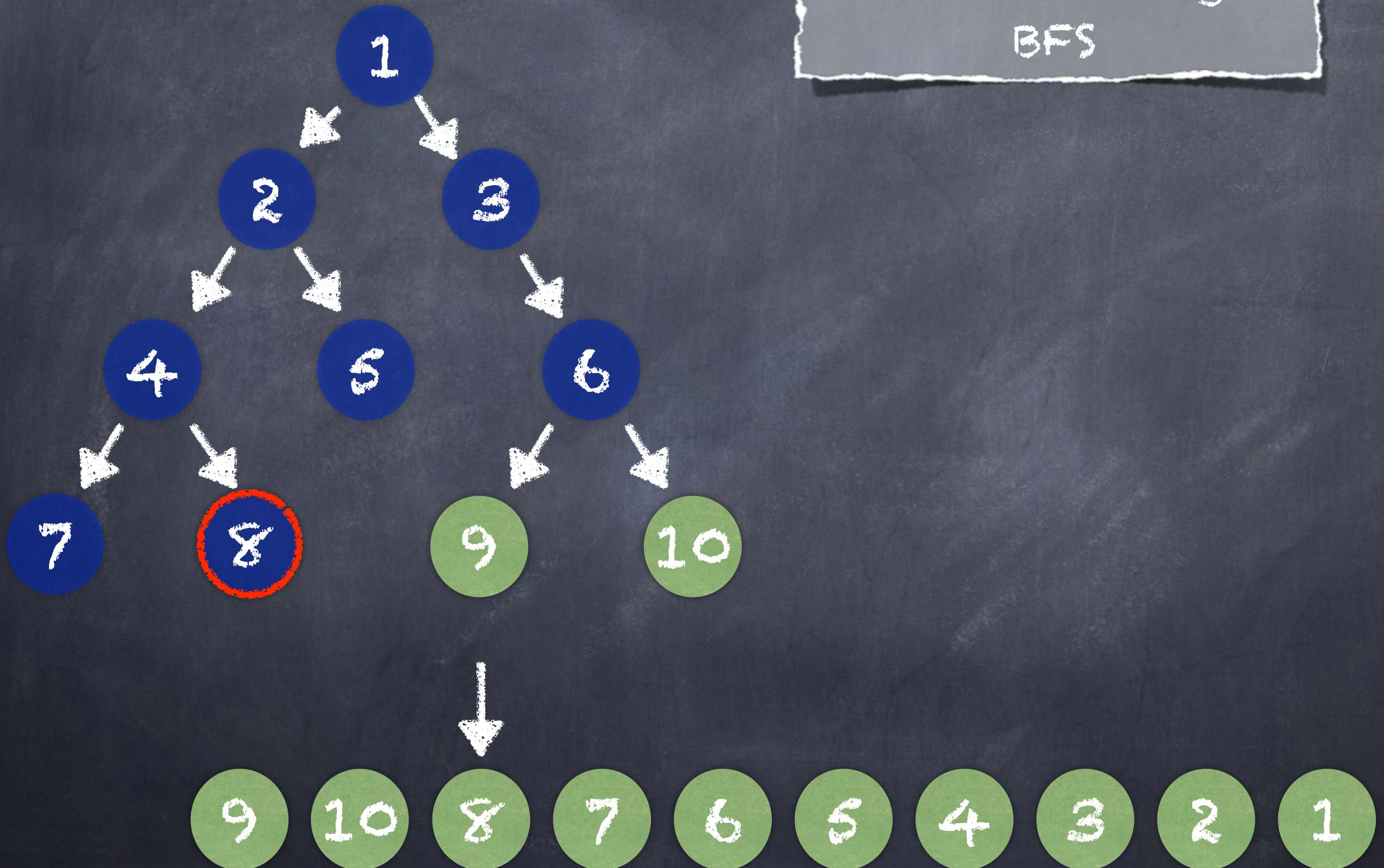
BFS



Pizzabestellung

Wiederholung

BFS



Pizzabestellung

Wiederholung

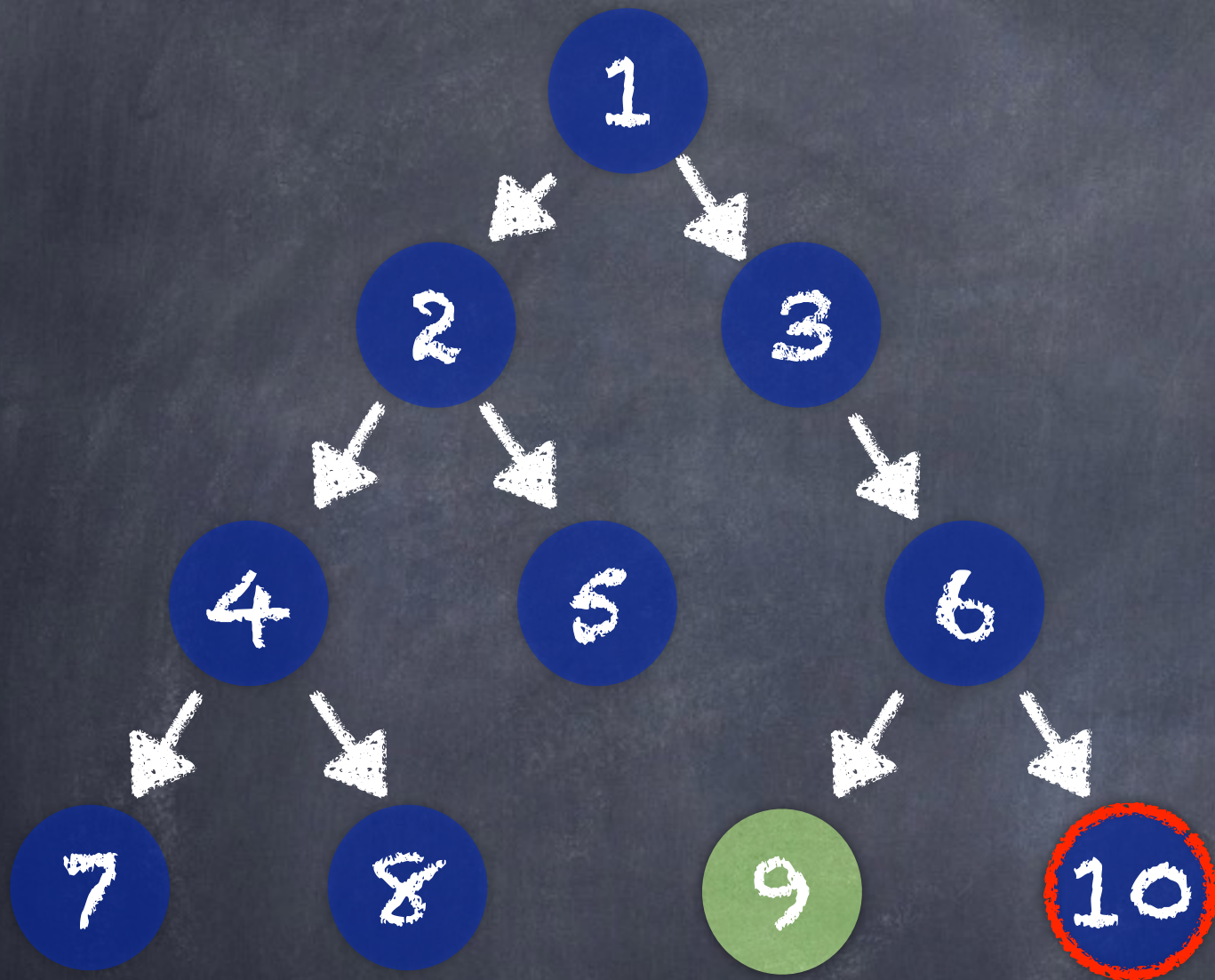
BFS



Pizzabestellung

Wiederholung

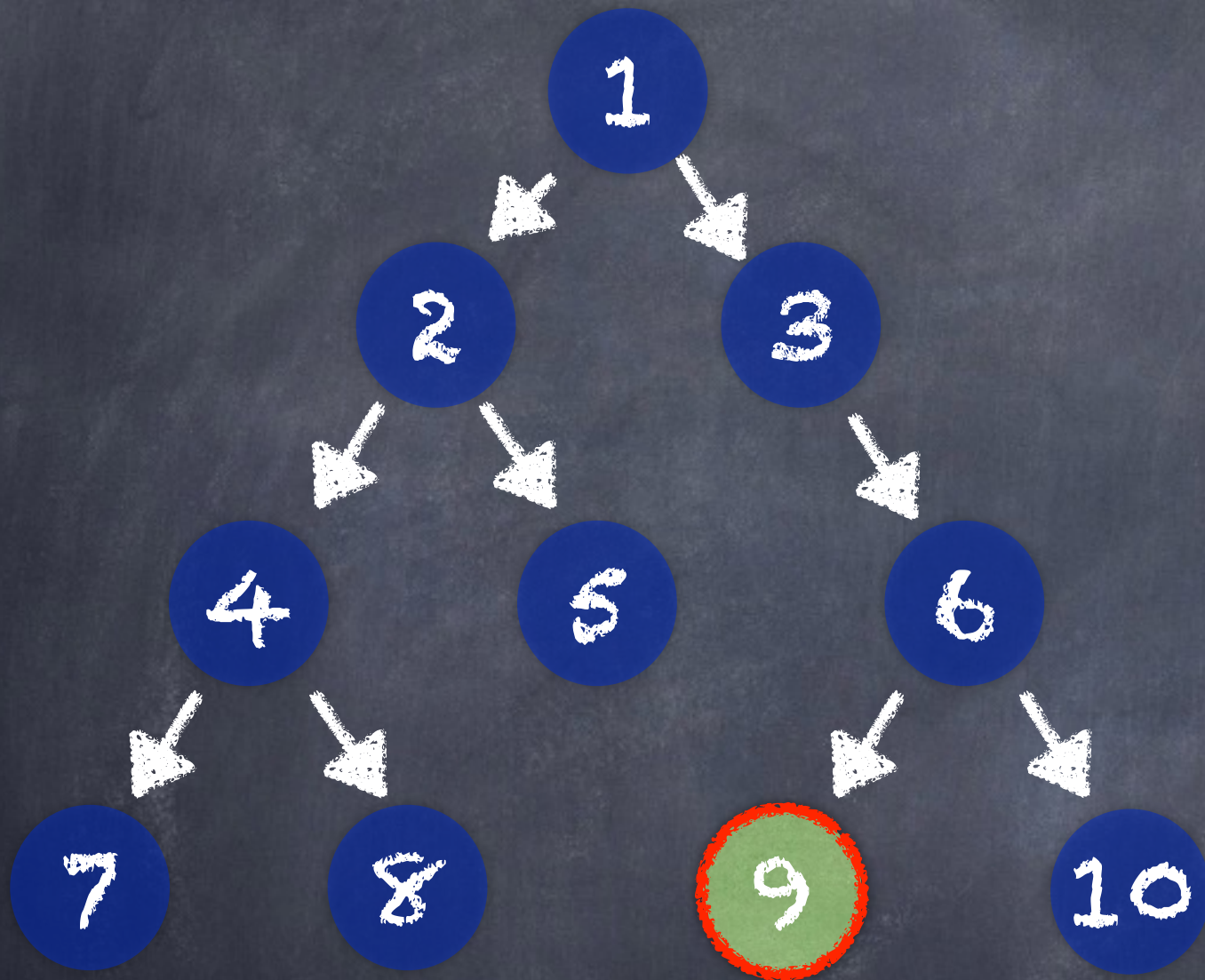
BFS



Pizzabestellung

Wiederholung

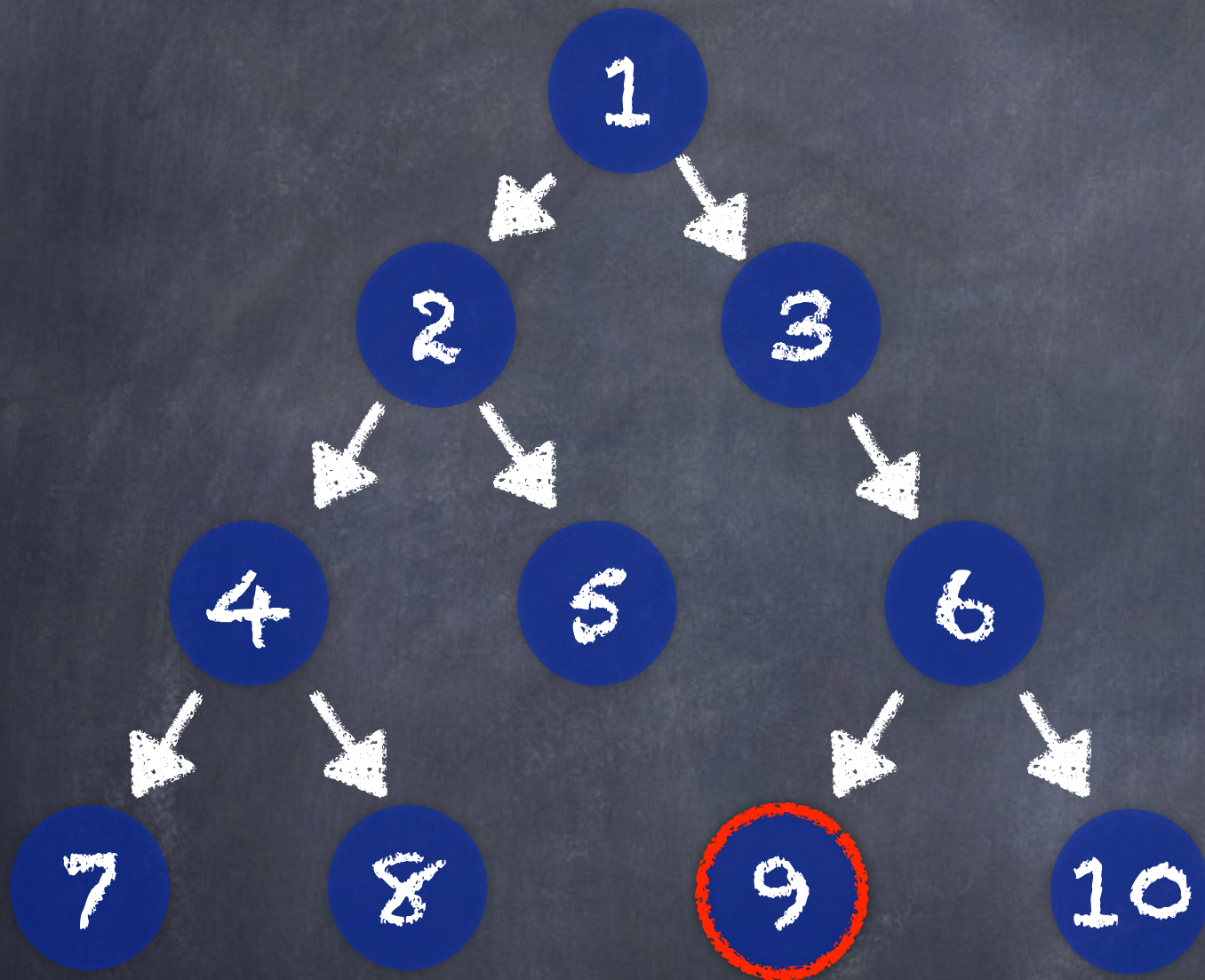
BFS



Pizzabestellung

Wiederholung

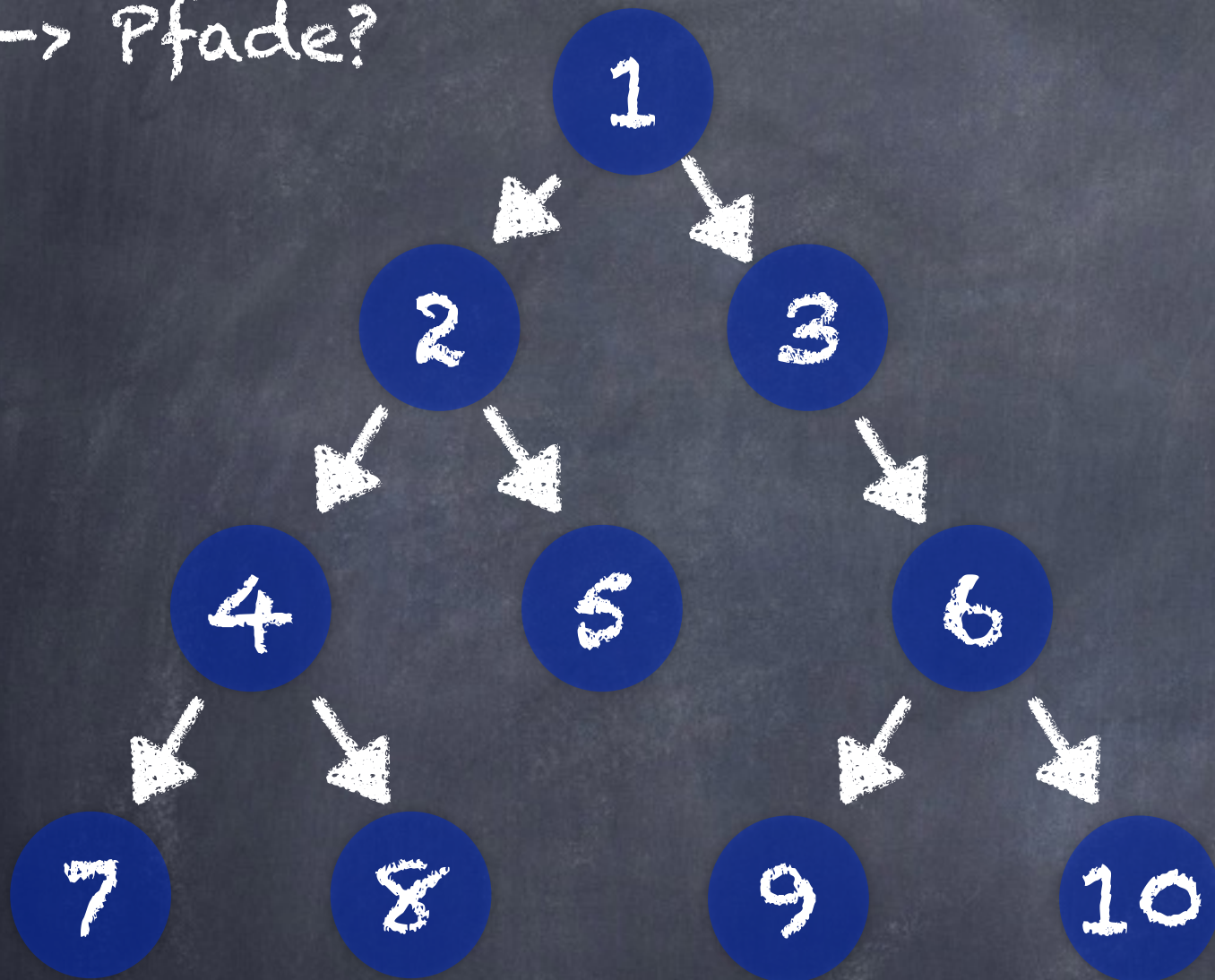
BFS



Pizzabestellung

Kanten gerichtet
→ Pfade?

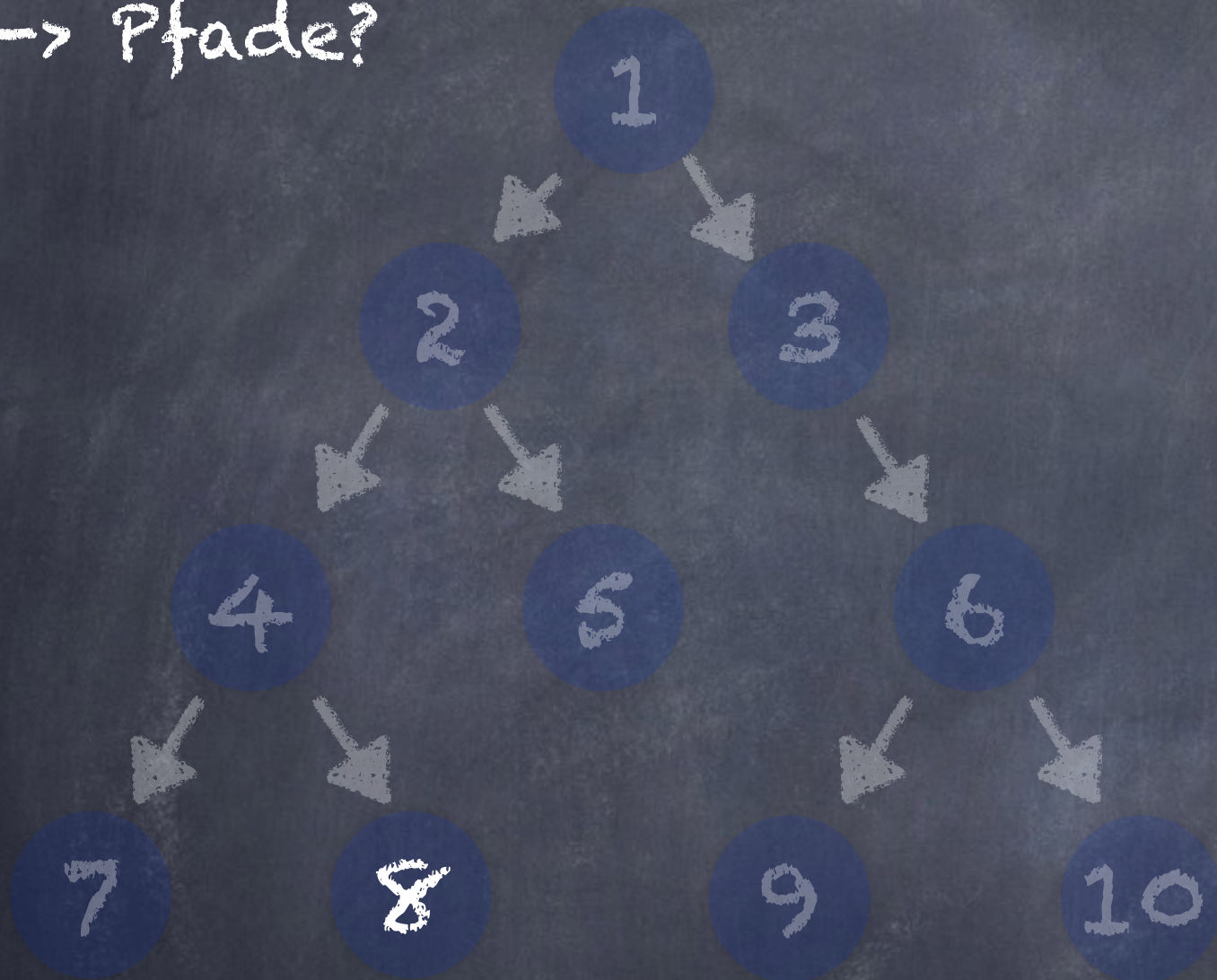
Wiederholung
BFS



Pizzabestellung

Kanten gerichtet
→ Pfade?

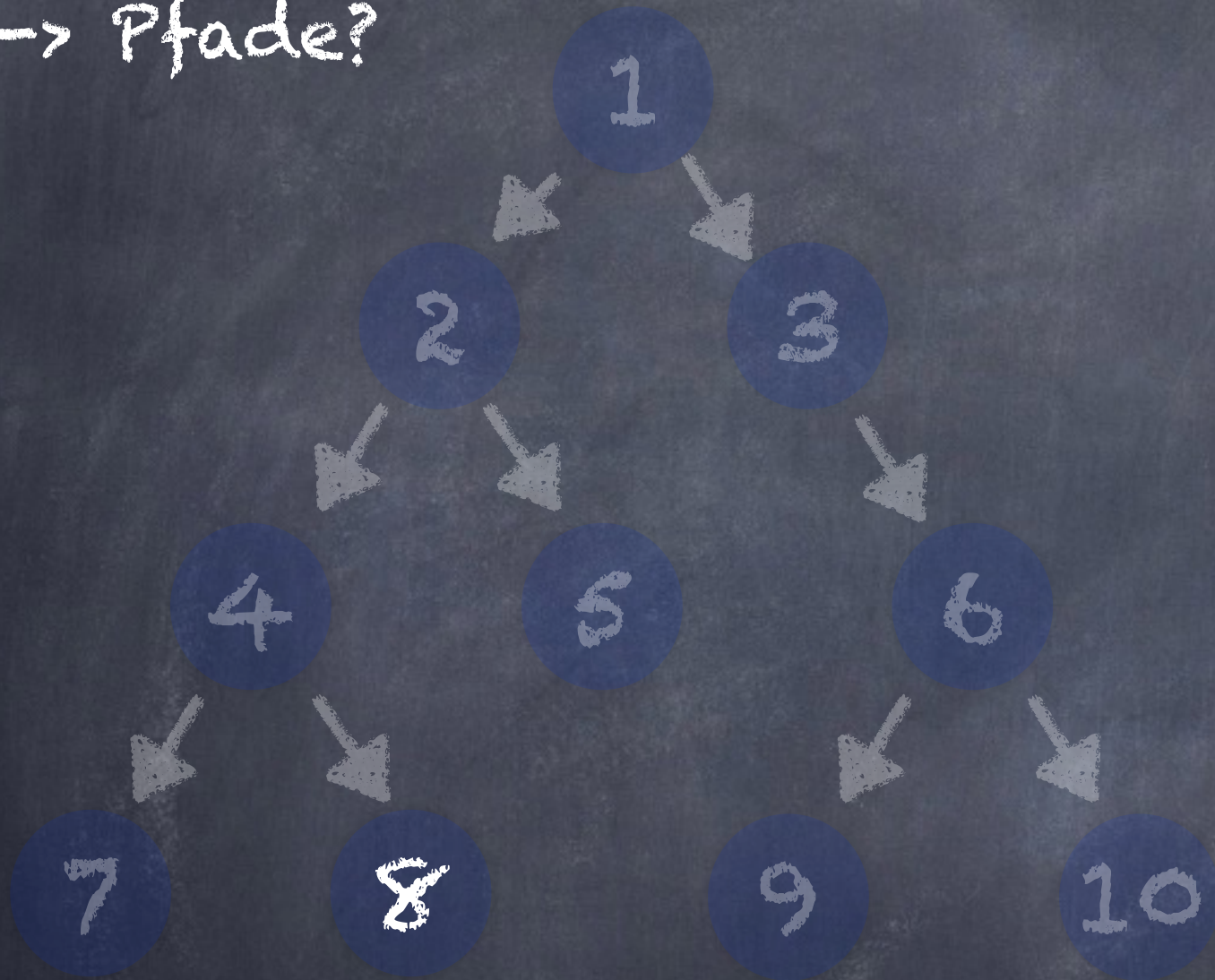
Wiederholung
BFS



Pizzabestellung

Kanten gerichtet
→ Pfade?

Wiederholung
BFS



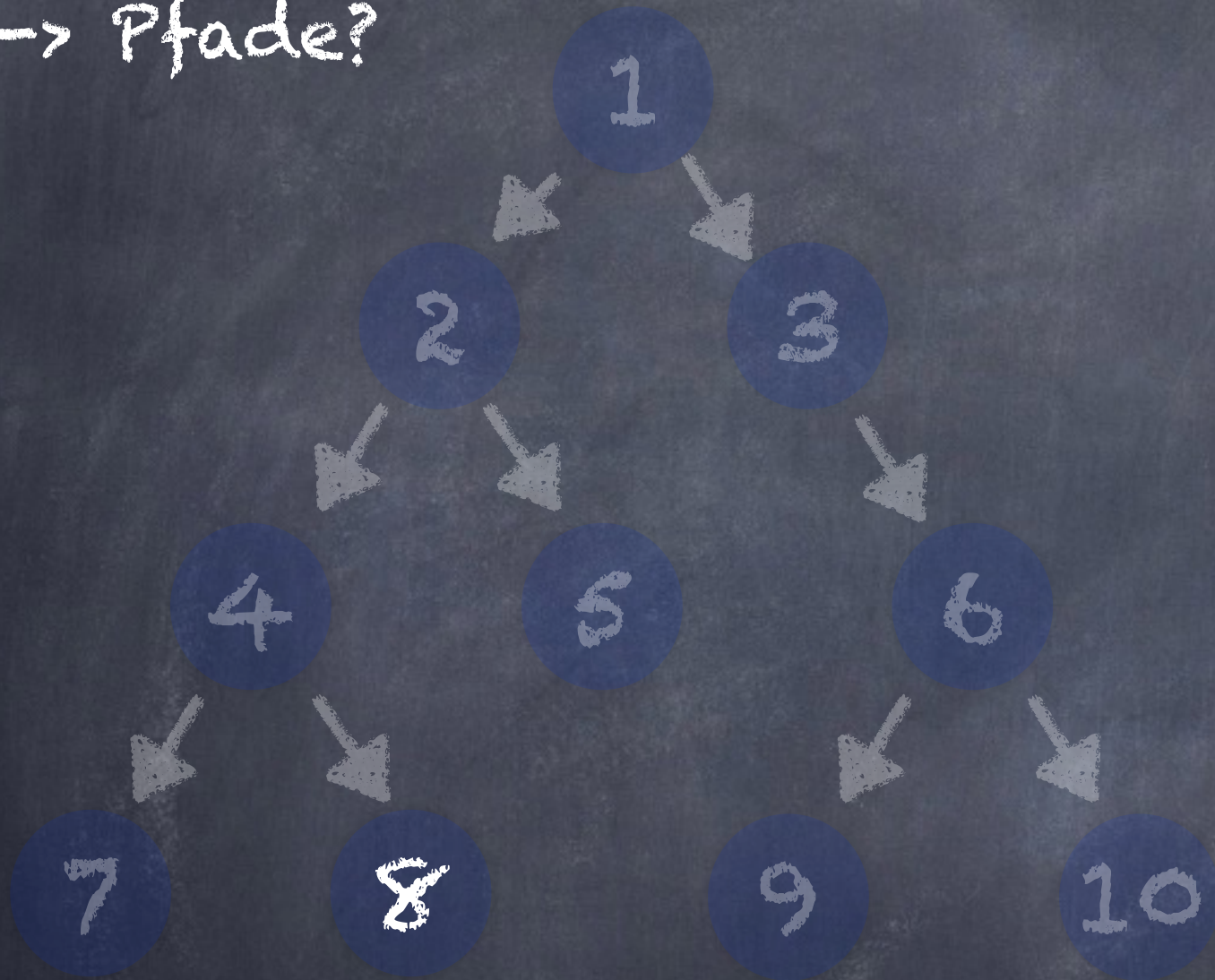
8 erreichbar
Aber wie?



Pizzabestellung

Kanten gerichtet
→ Pfade?

Wiederholung
BFS



8 erreichbar → Pfad zwischen 8 und 1

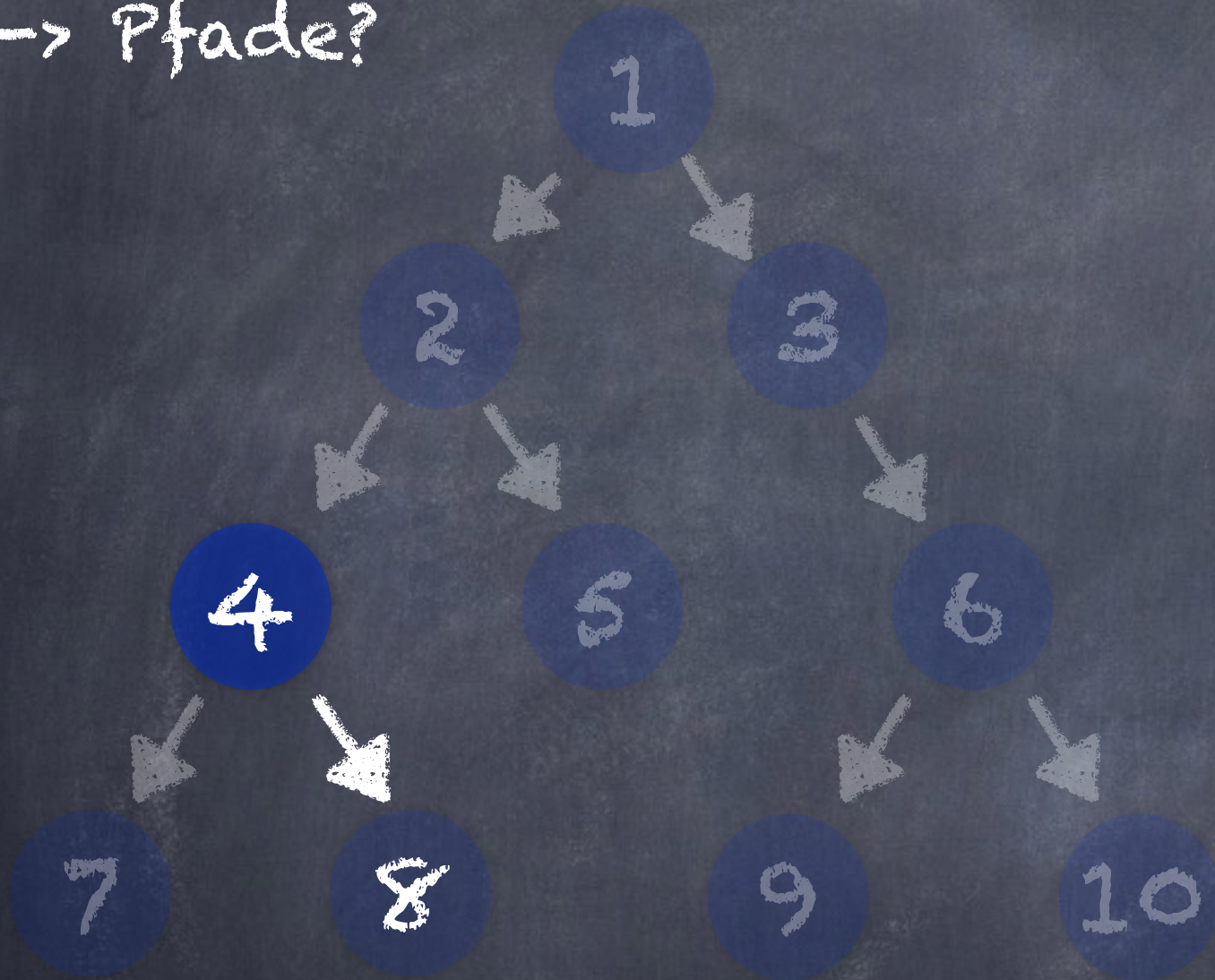
Aber wie?



Pizzabestellung

Kanten gerichtet
→ Pfade?

Wiederholung
BFS



8 erreichbar → Pfad zwischen 8 und 1

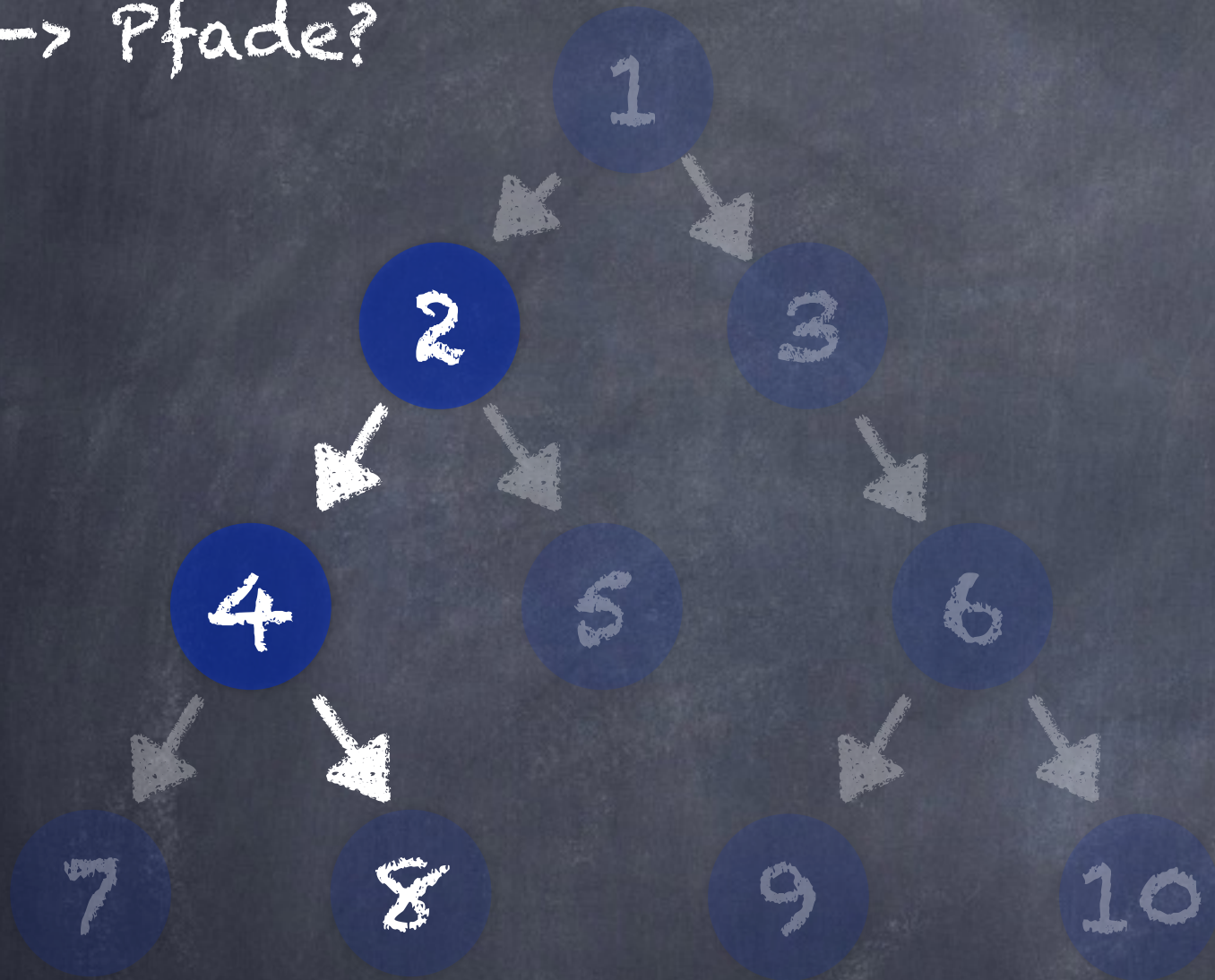
Aber wie?



Pizzabestellung

Kanten gerichtet
→ Pfade?

Wiederholung
BFS



8 erreichbar → Pfad zwischen 8 und 1

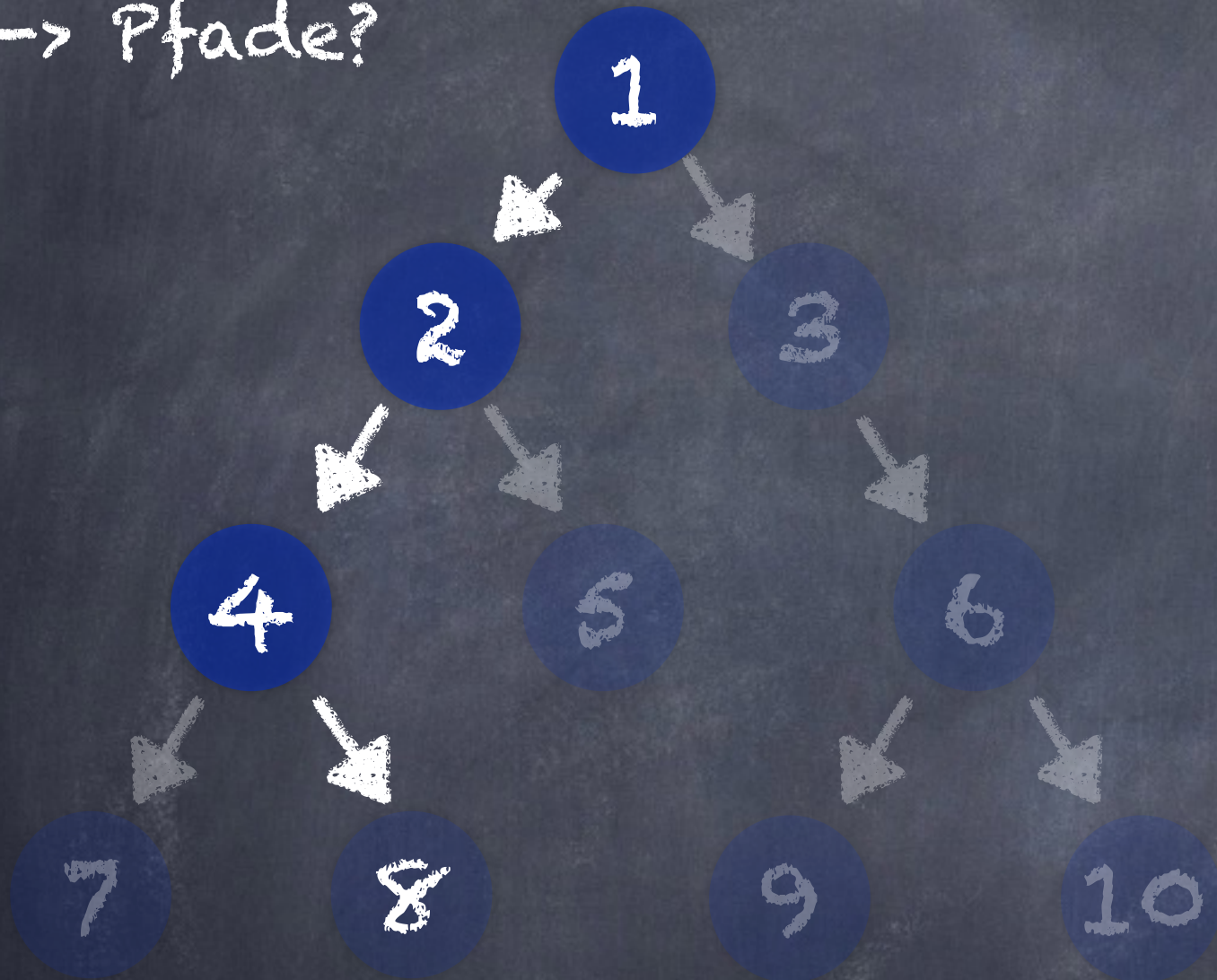
Aber wie?



Pizzabestellung

Kanten gerichtet
→ Pfade?

Wiederholung
BFS



8 erreichbar → Pfad zwischen 8 und 1

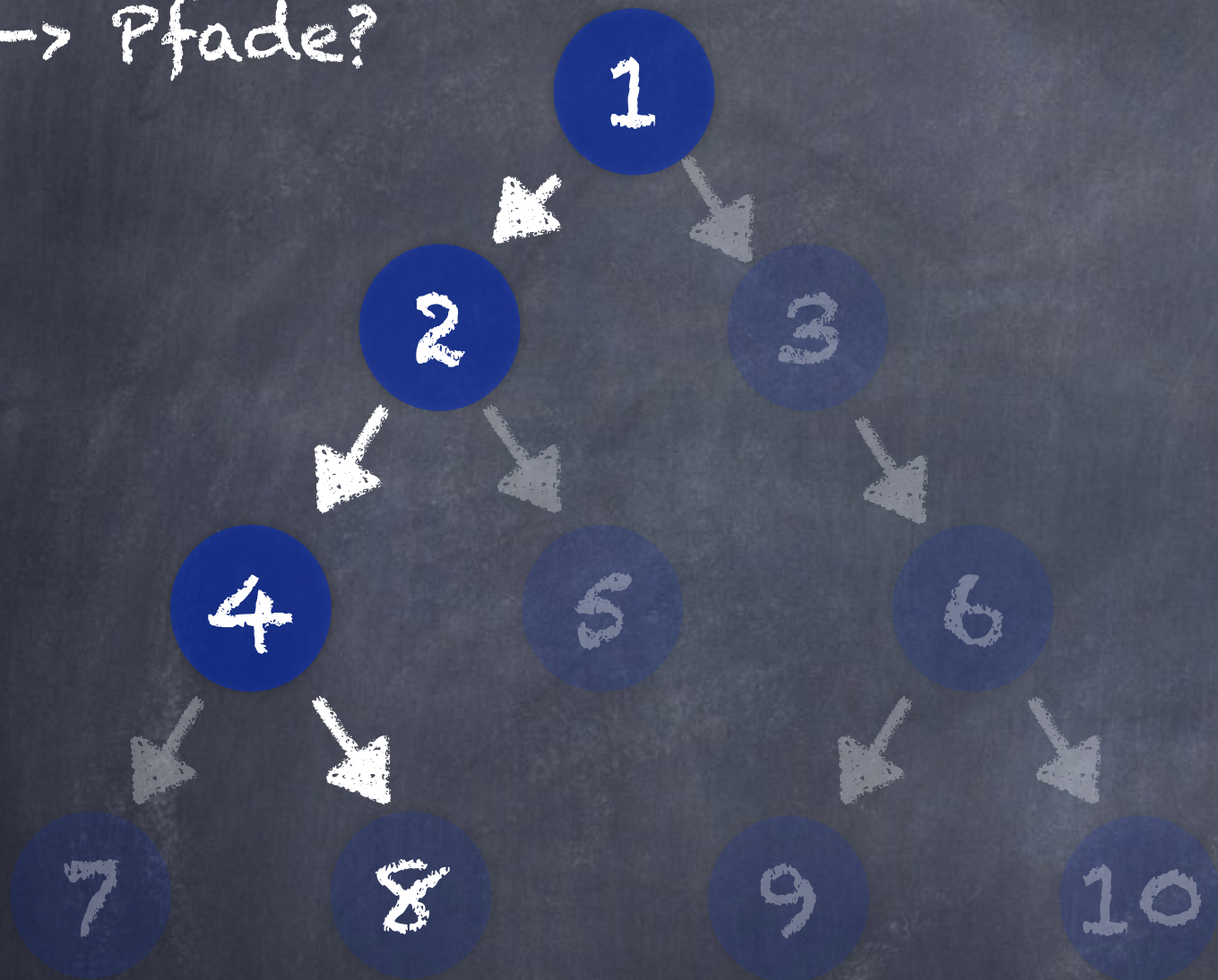
Aber wie?



Pizzabestellung

Kanten gerichtet
→ Pfade?

Wiederholung
BFS



v erreichbar → Pfad zwischen v und 1

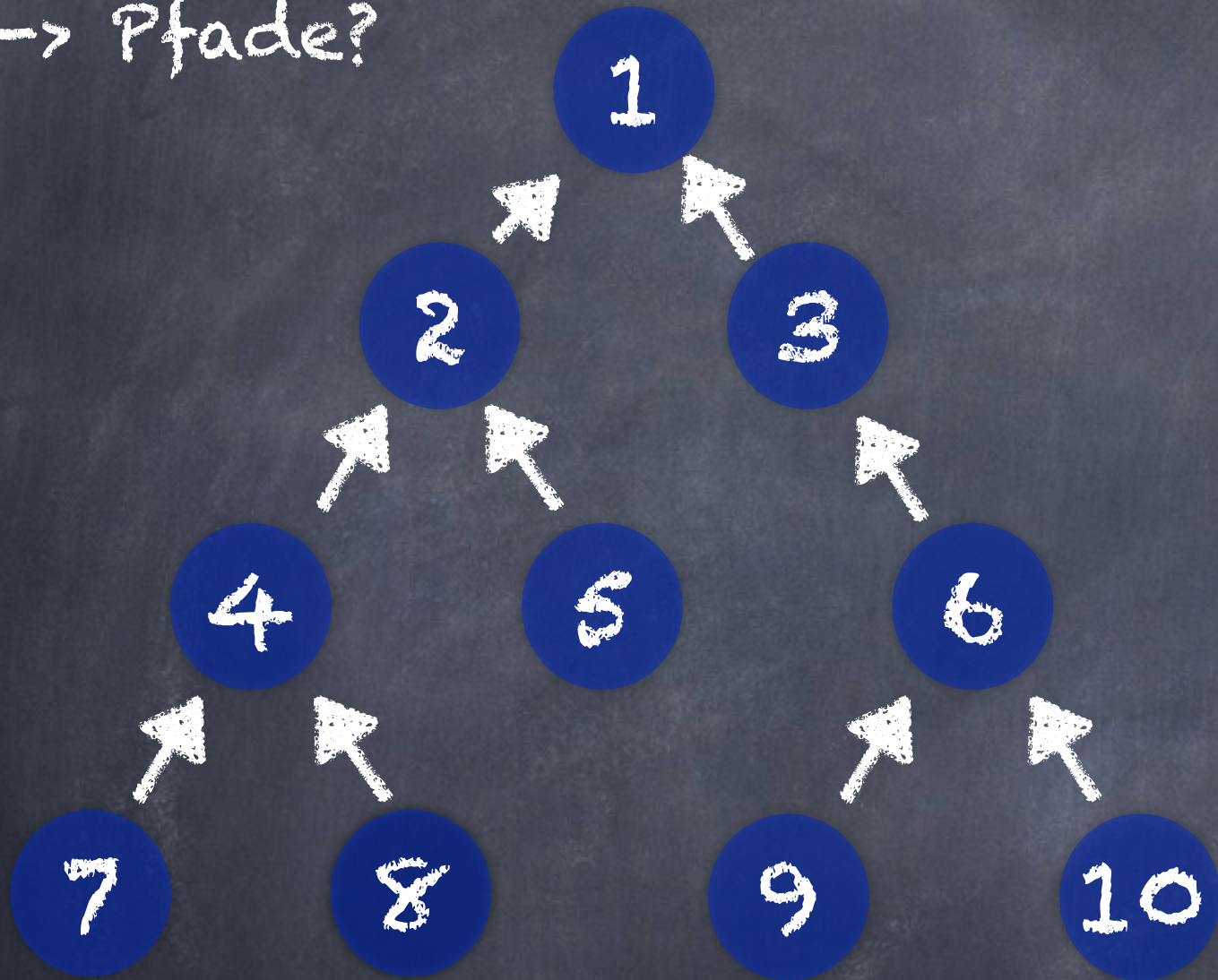
Aber wie?



Pizzabestellung

Kanten gerichtet
→ Pfade?

Wiederholung
BFS



v erreichbar → Pfad zwischen v und 1

Aber wie?

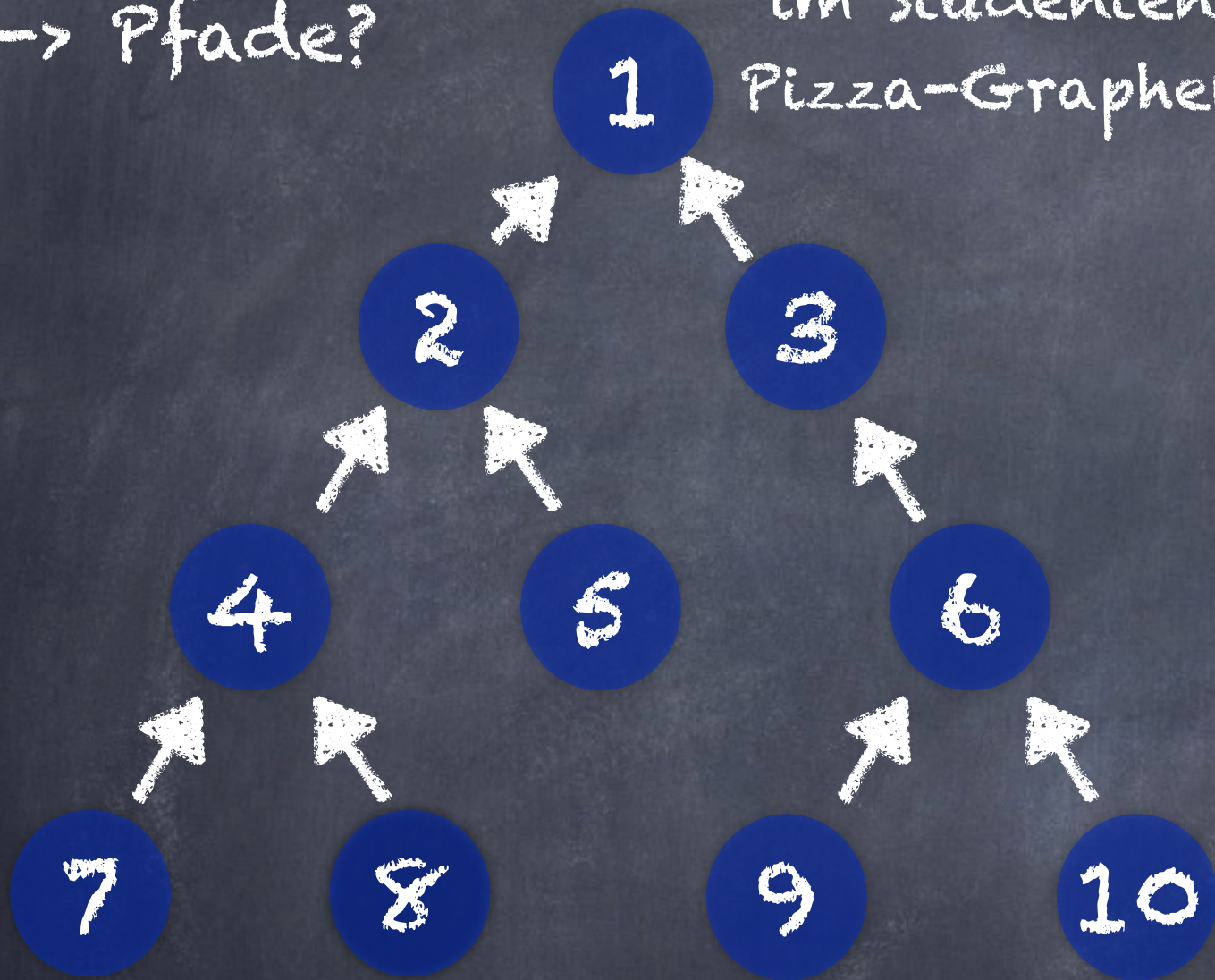


Pizzabestellung

Kanten gerichtet
→ Pfade?

Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung
BFS



v erreichbar → Pfad zwischen v und 1

Aber wie?



Pizzabestellung

Kanten gerichtet
→ Pfade?

1

Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung

BFS

v erreichbar → Pfad zwischen v und 1

Aber wie?

Pizzabestellung

Kanten gerichtet
→ Pfade?



Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung
BFS

v erreichbar → Pfad zwischen v und 1
Aber wie?

Pizzabestellung

Kanten gerichtet
→ Pfade?



Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung
BFS



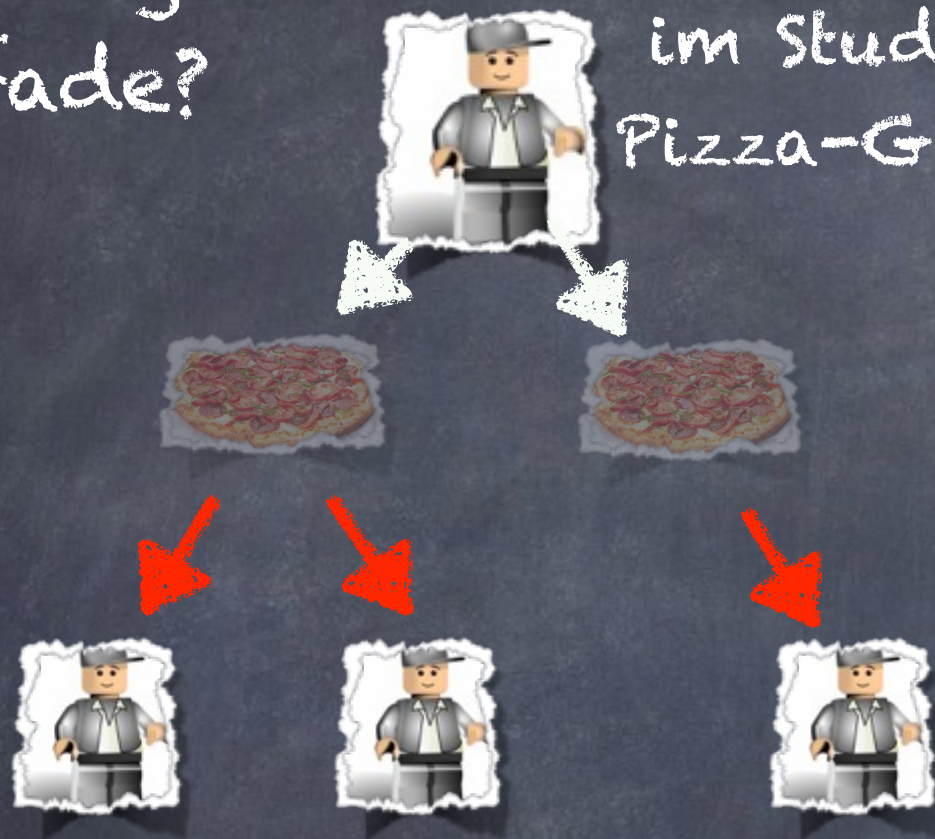
v erreichbar → Pfad zwischen v und 1
Aber wie?

Pizzabestellung

Kanten gerichtet
→ Pfade?

Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung
BFS



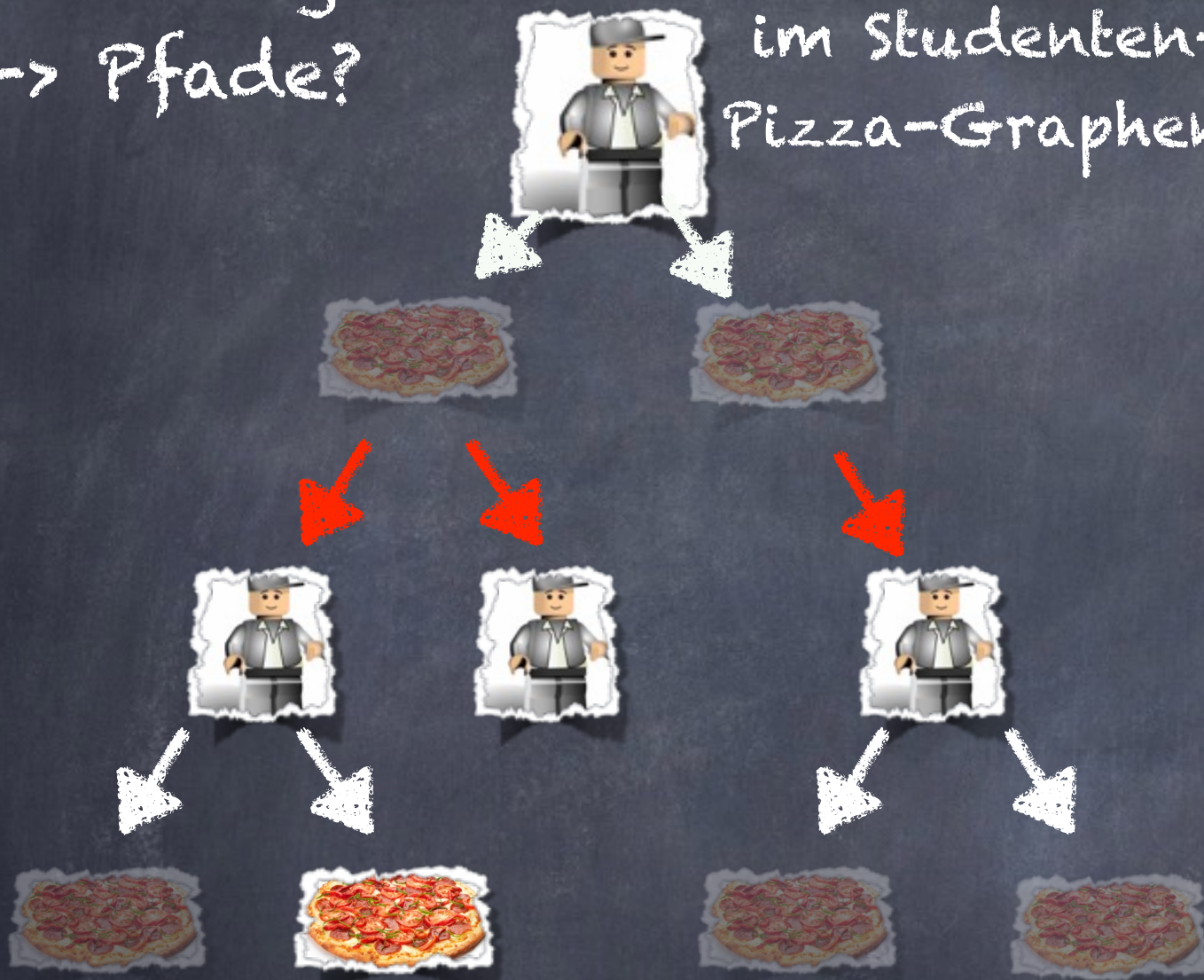
v erreichbar → Pfad zwischen v und 1
Aber wie?

Pizzabestellung

Kanten gerichtet
→ Pfade?

Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung
BFS



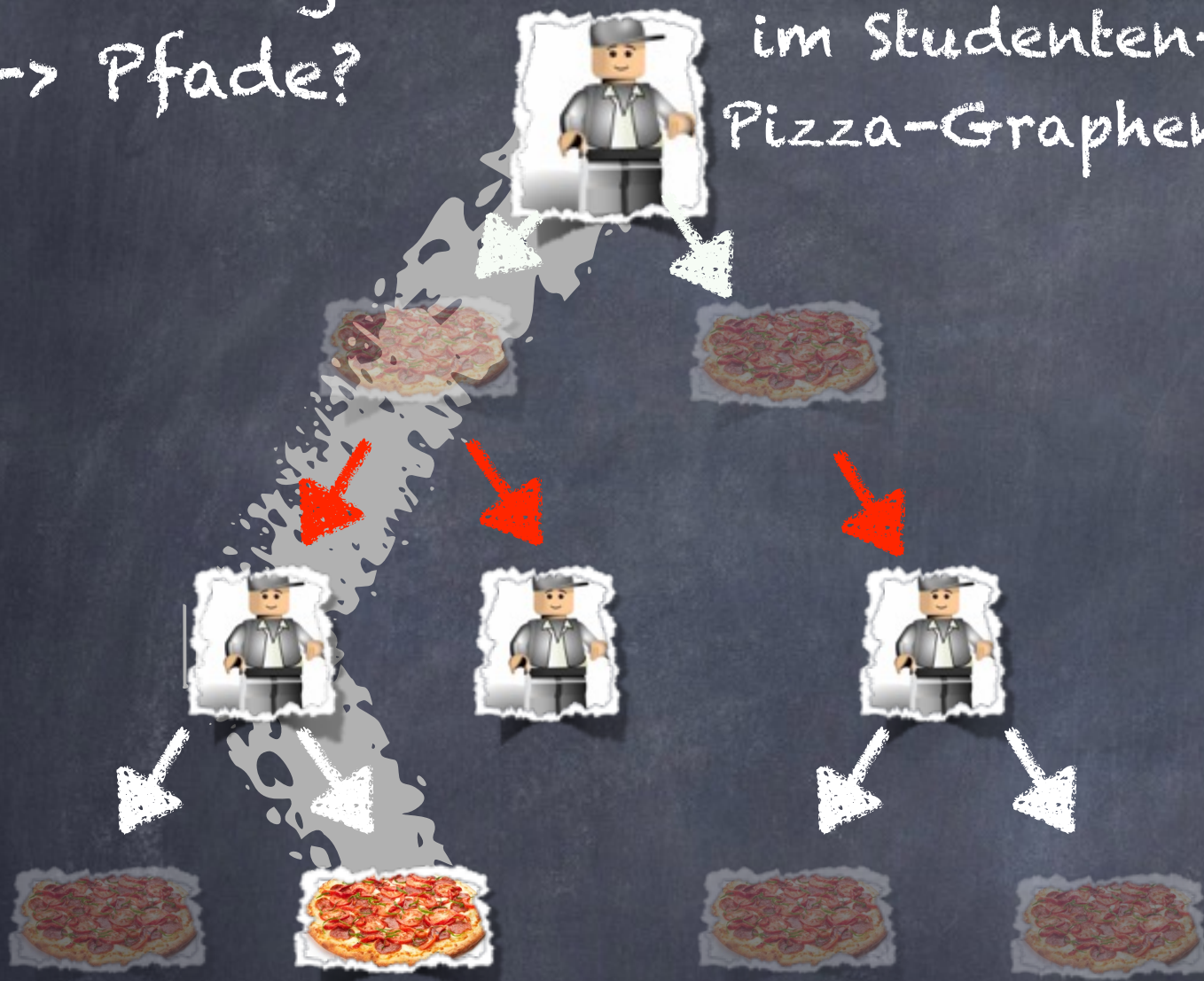
v erreichbar → Pfad zwischen v und 1
Aber wie?

Pizzabestellung

Kanten gerichtet
→ Pfade?

Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung
BFS



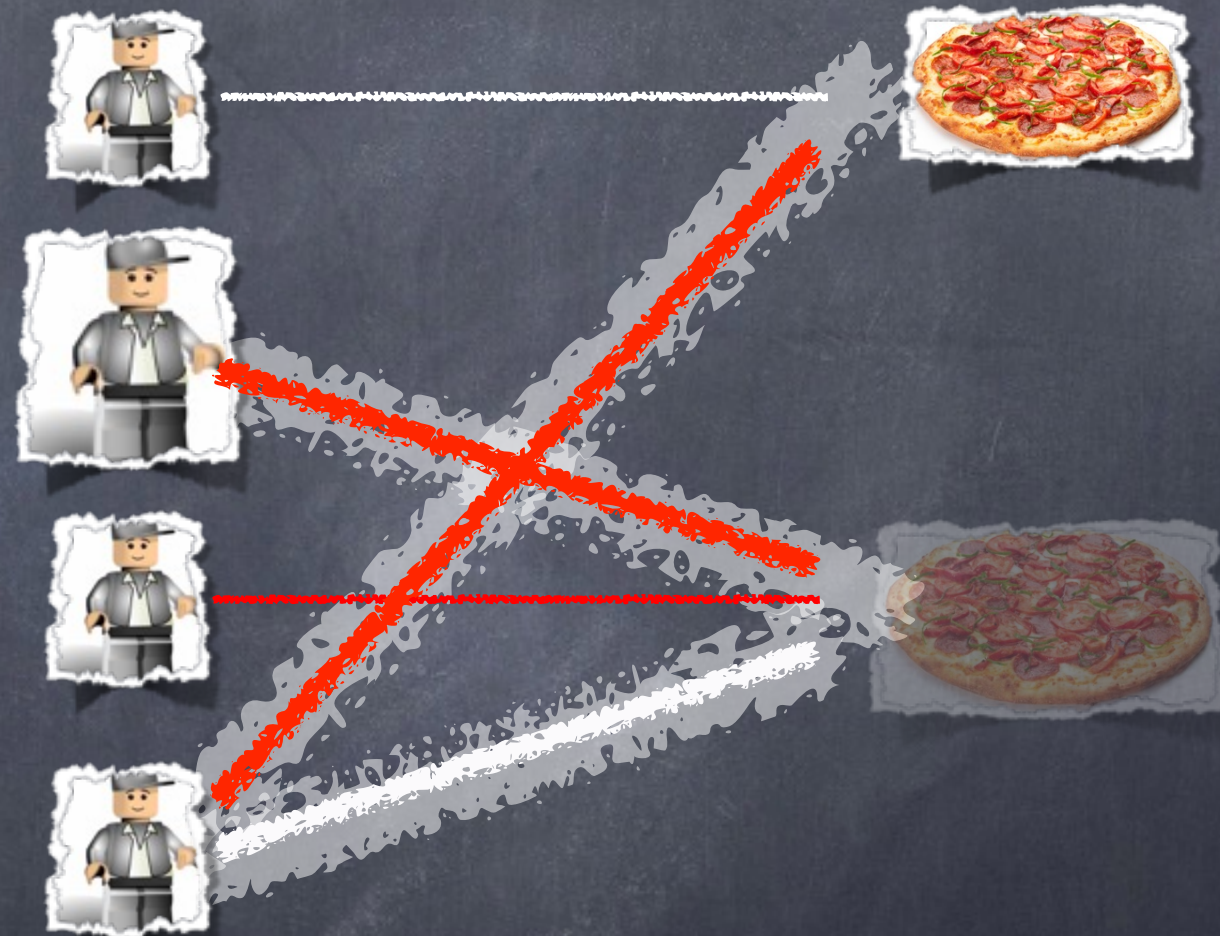
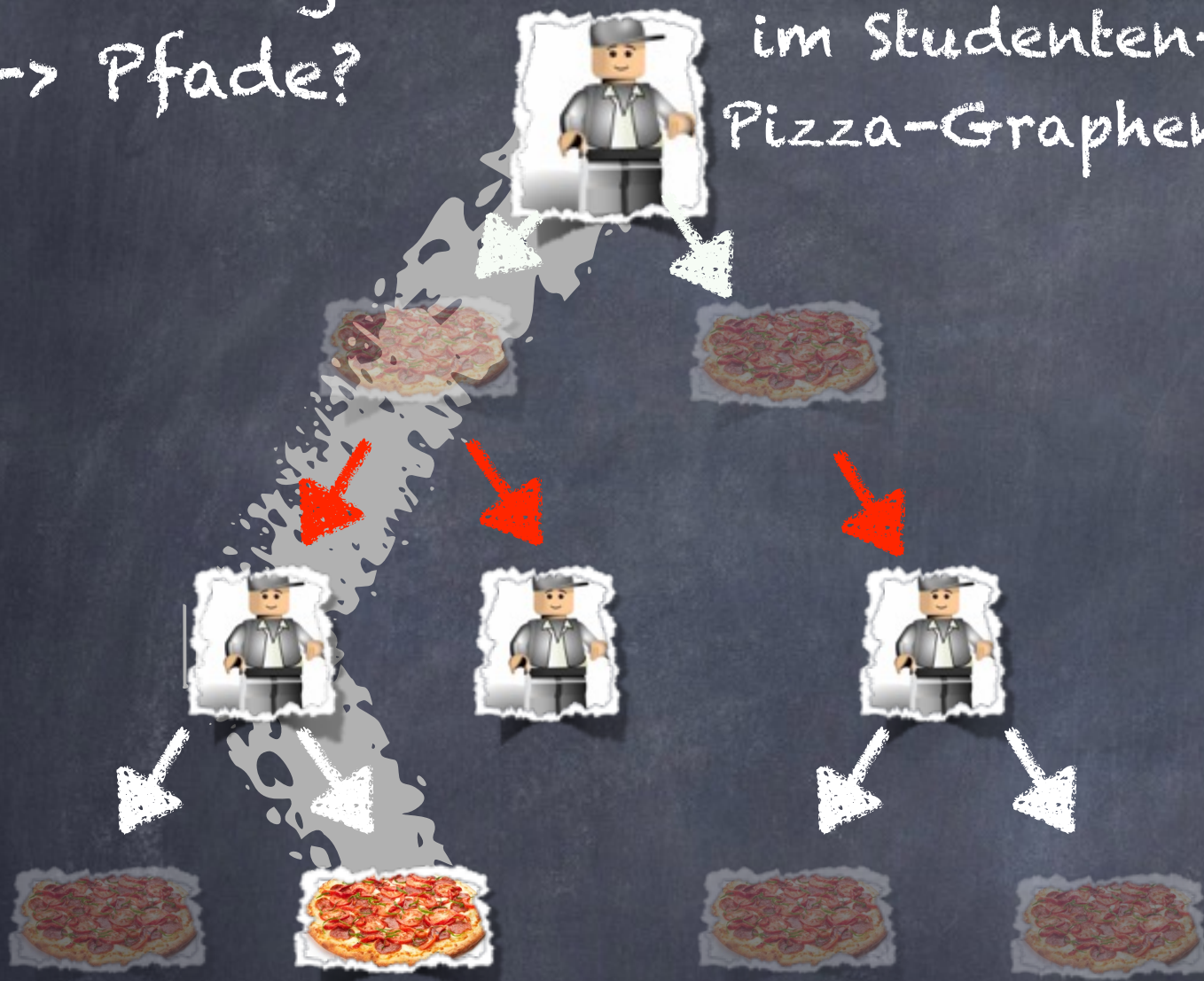
v erreichbar → Pfad zwischen v und 1
Aber wie?

Pizzabestellung

Kanten gerichtet
→ Pfade?

Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung
BFS



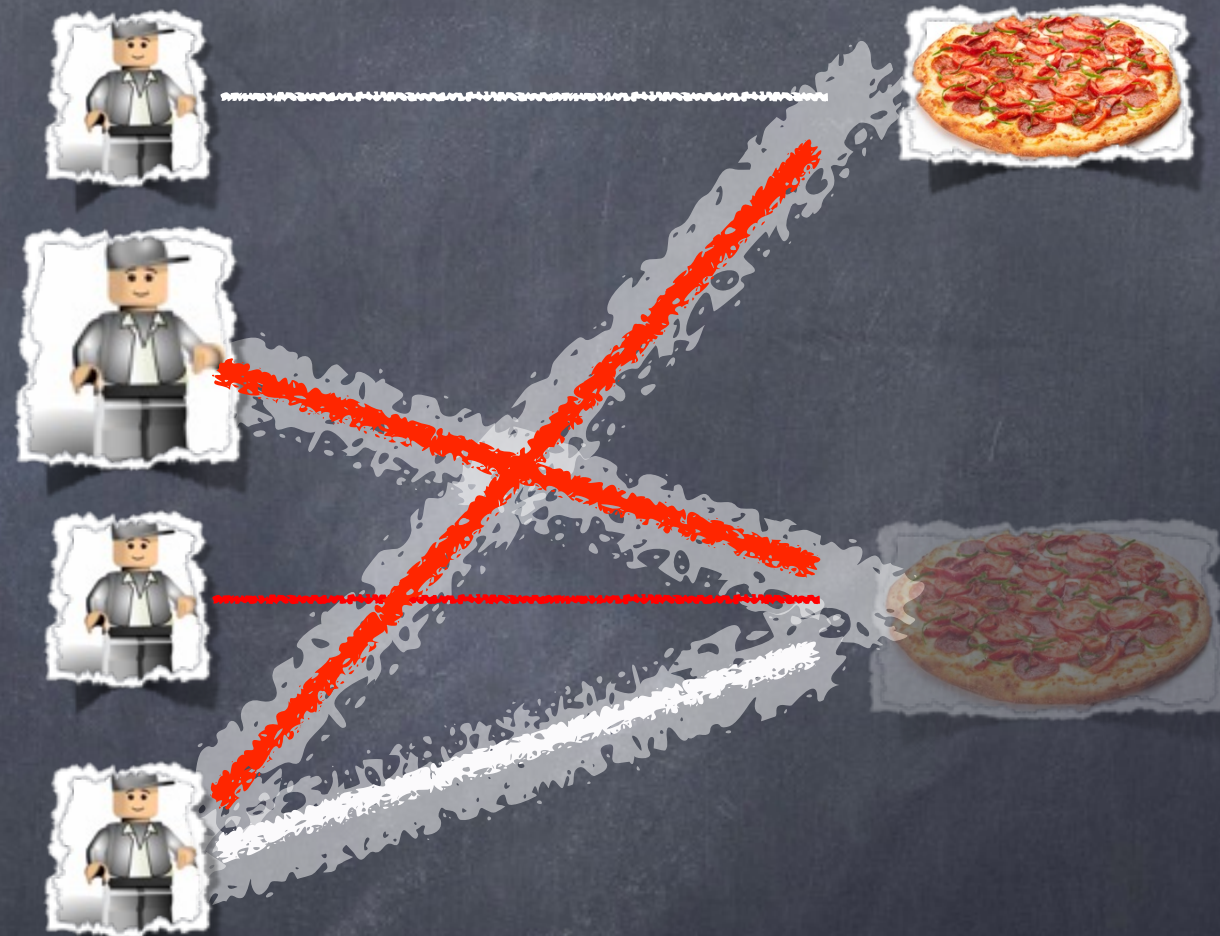
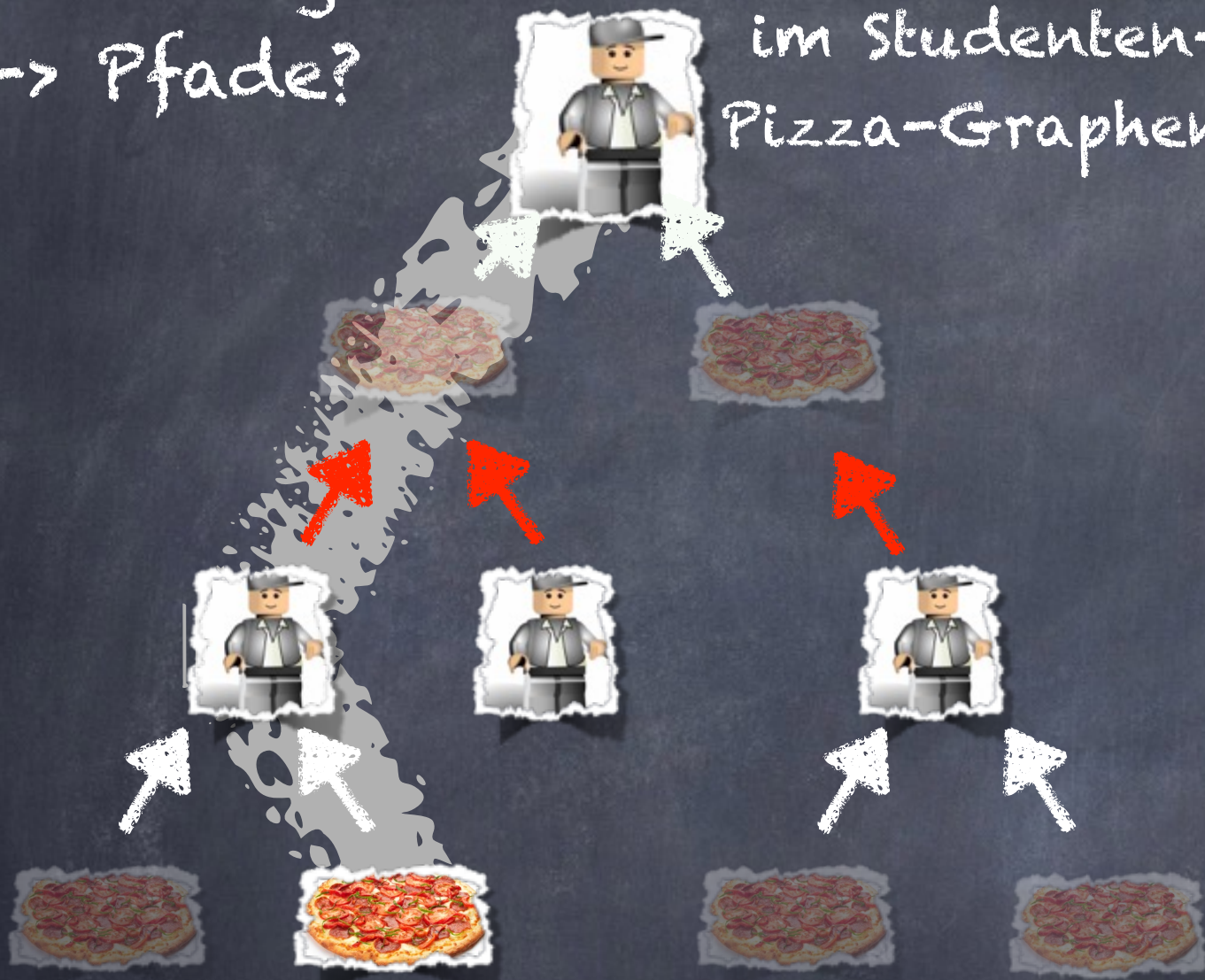
v erreichbar → Pfad zwischen v und 1
Aber wie?

Pizzabestellung

Kanten gerichtet
→ Pfade?

Was heißt das
im Studenten-
Pizza-Graphen?

Wiederholung
BFS



v erreichbar → Pfad zwischen v und 1
Aber wie?

Pizzabestellung

Wiederholung
DFS



Pizzabestellung

Wiederholung
DFS

DFS = depth-first search



Pizzabestellung

Wiederholung
DFS

DFS = depth-first search
(Tiefensuche)



Pizzabestellung

Wiederholung
DFS

DFS = depth-first search
(Tiefensuche)

verw. Datenstruktur:



Pizzabestellung

Wiederholung
DFS

DFS = depth-first search
(Tiefensuche)

verw. Datenstruktur:
Stack (Stapel)



Pizzabestellung

Wiederholung
DFS

DFS = depth-first search
(Tiefensuche)

verw. Datenstruktur:

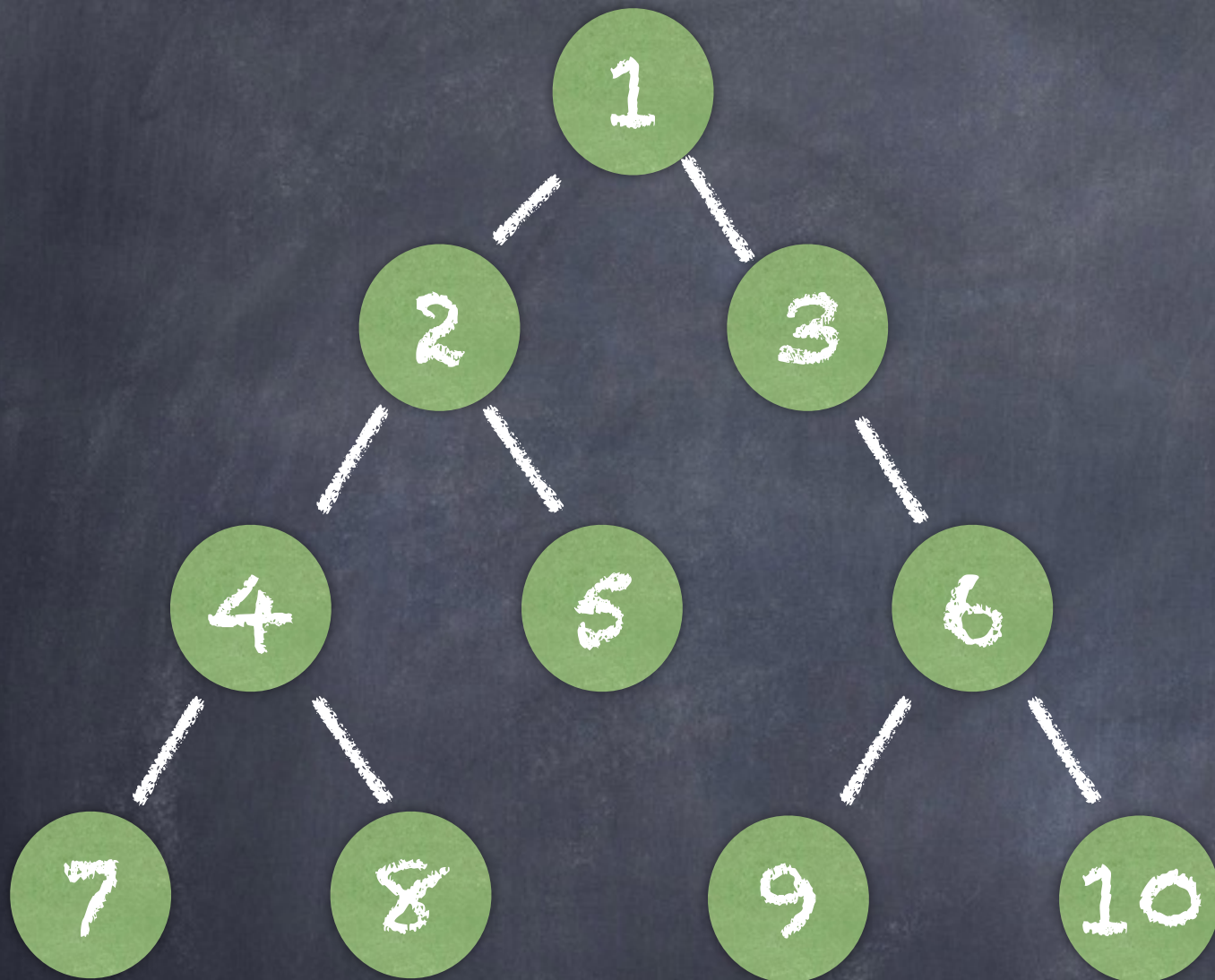
Stack (Stapel)

last-in-first-out (Lifo)



Pizzabestellung

Wiederholung
DFS



DFS = depth-first search
(Tiefensuche)

verw. Datenstruktur:

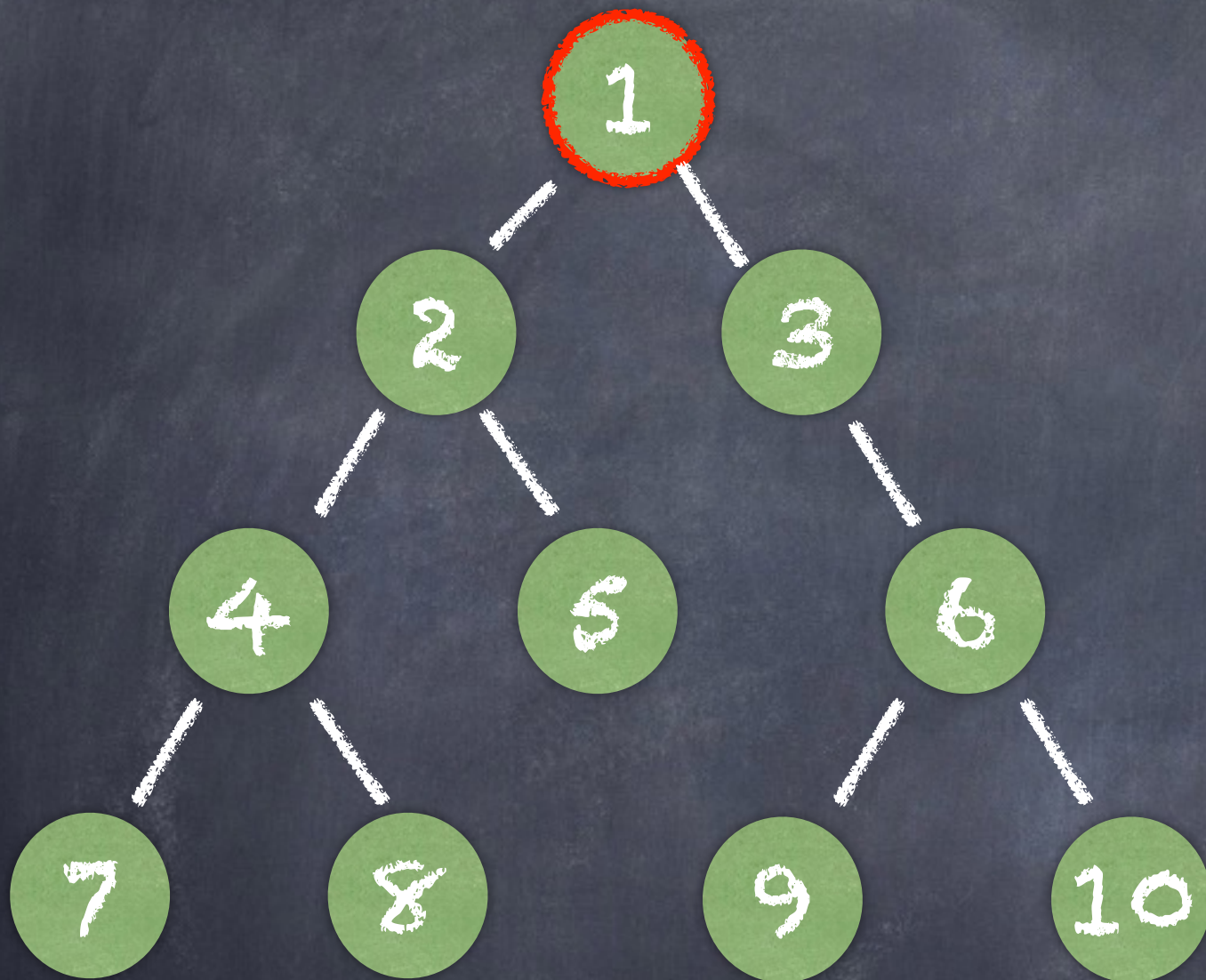
Stack (Stapel)

last-in-first-out (Lifo)



Pizzabestellung

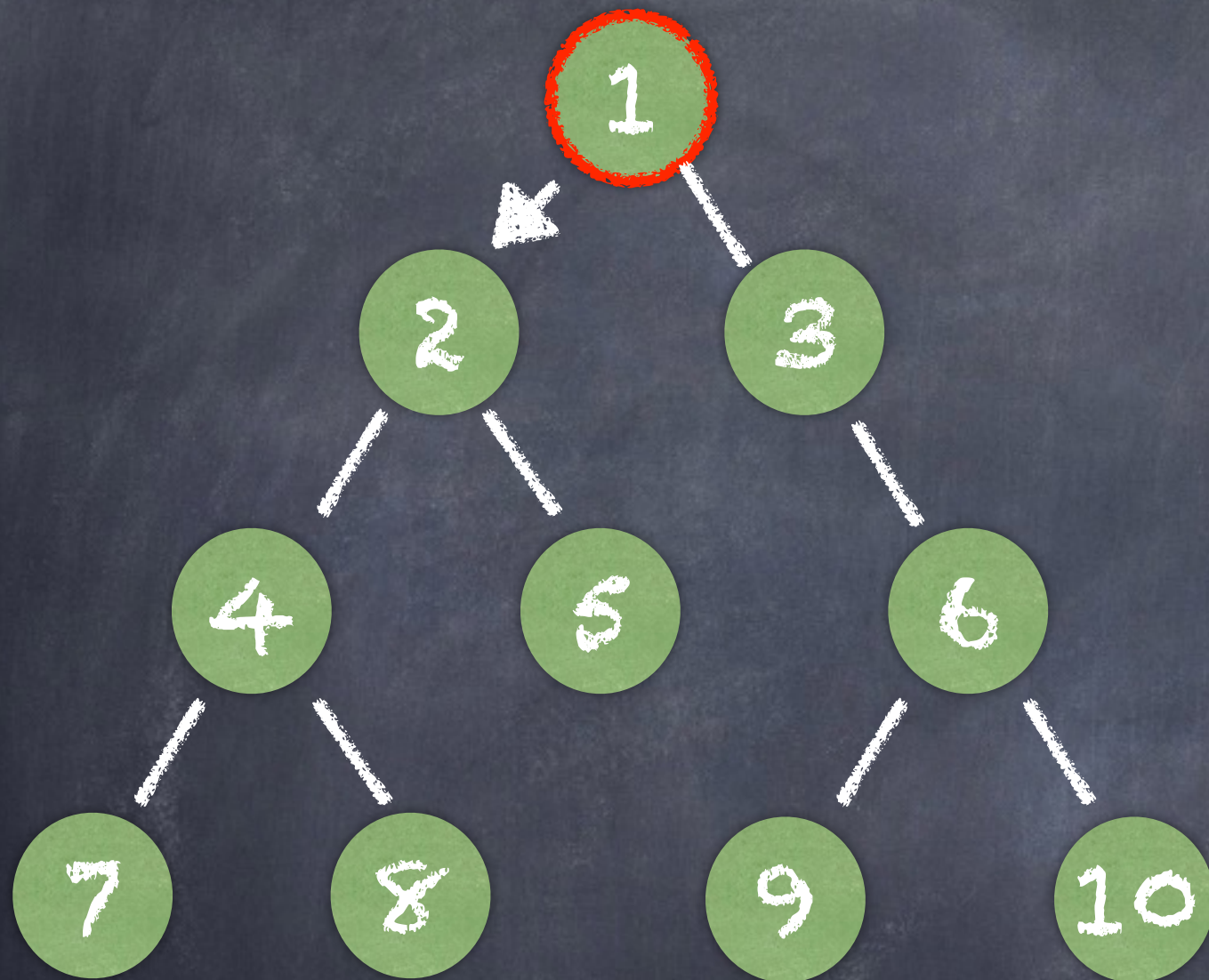
Wiederholung
DFS



1

Pizzabestellung

Wiederholung
DFS

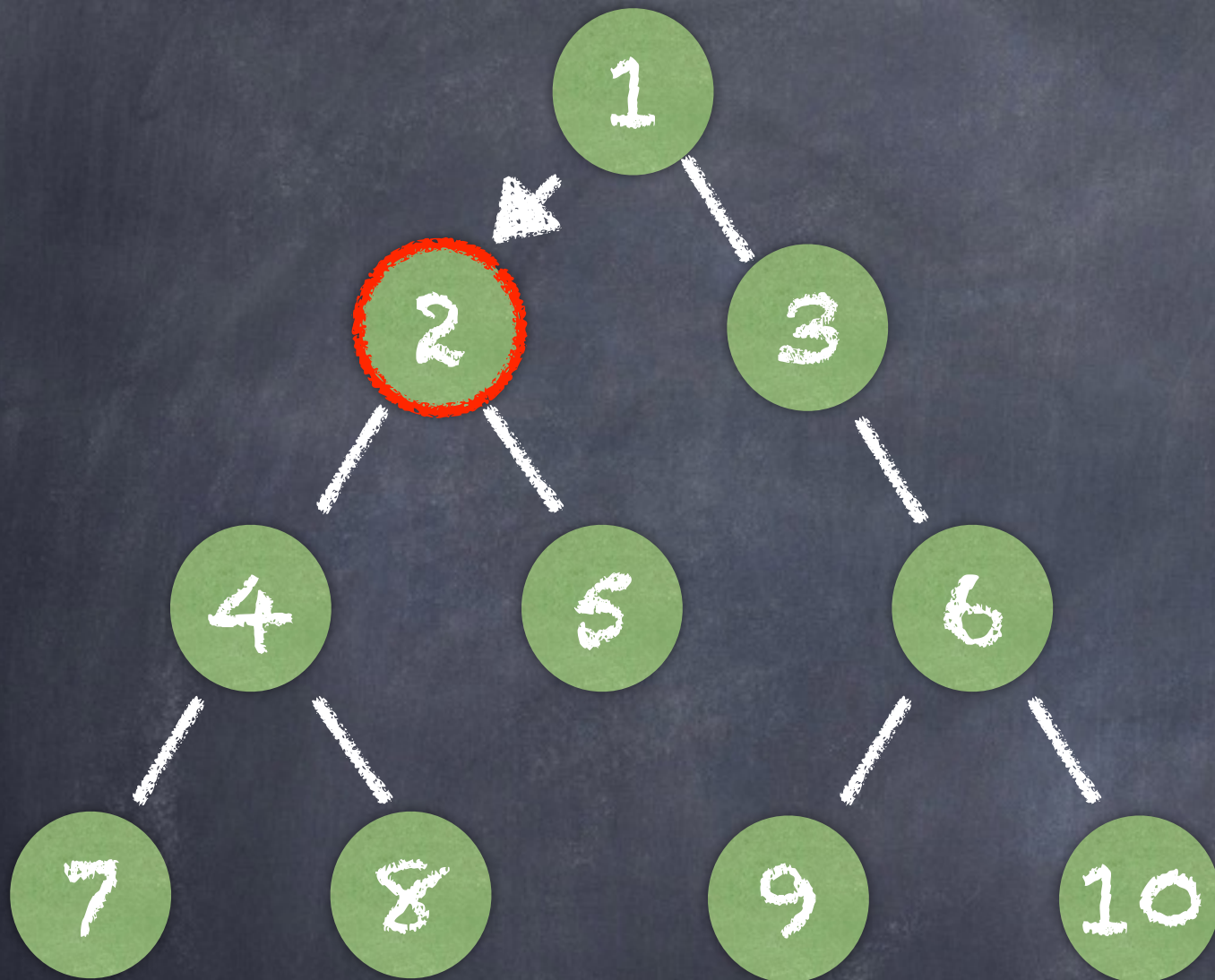


2

1

Pizzabestellung

Wiederholung
DFS

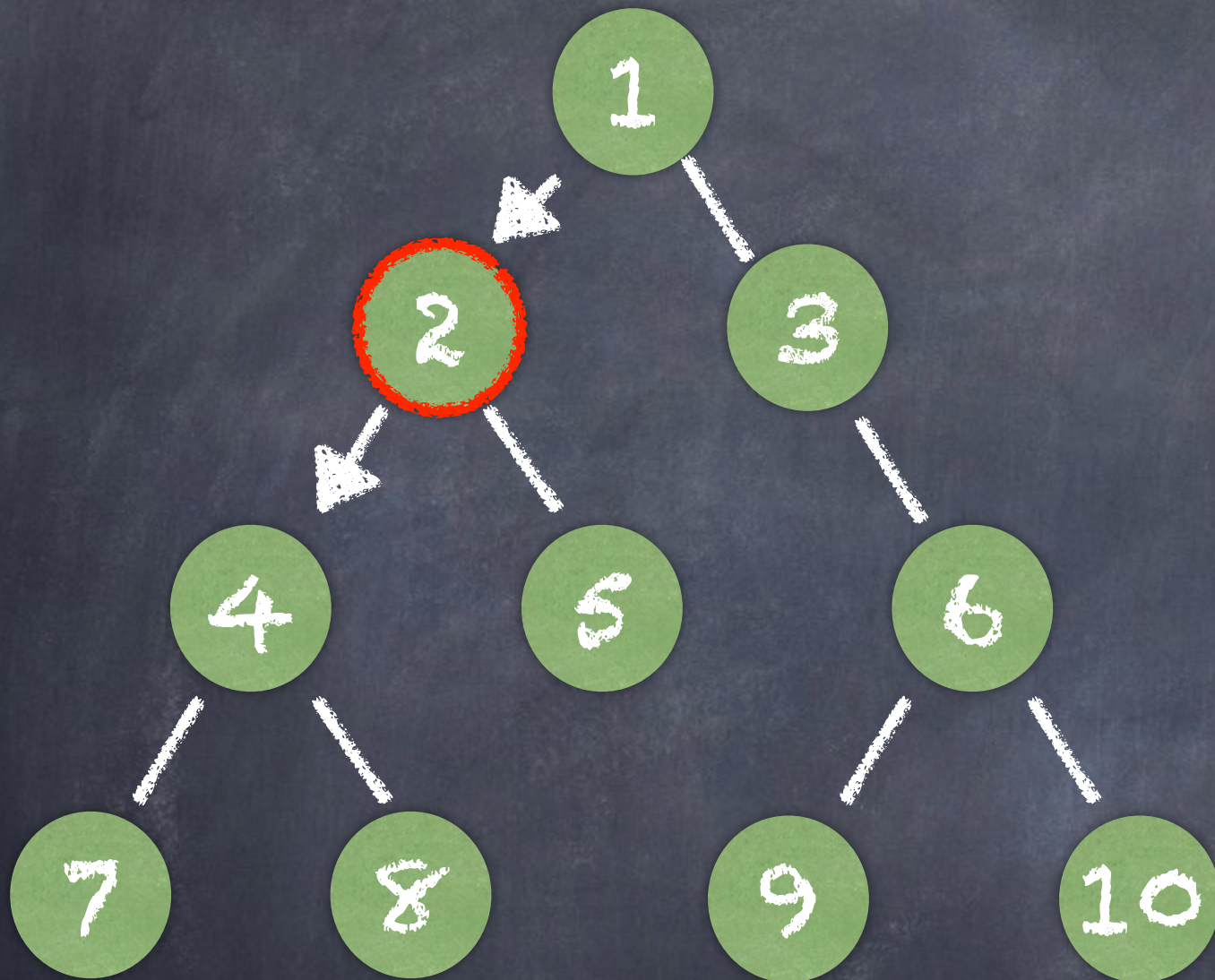


2

1

Pizzabestellung

Wiederholung
DFS



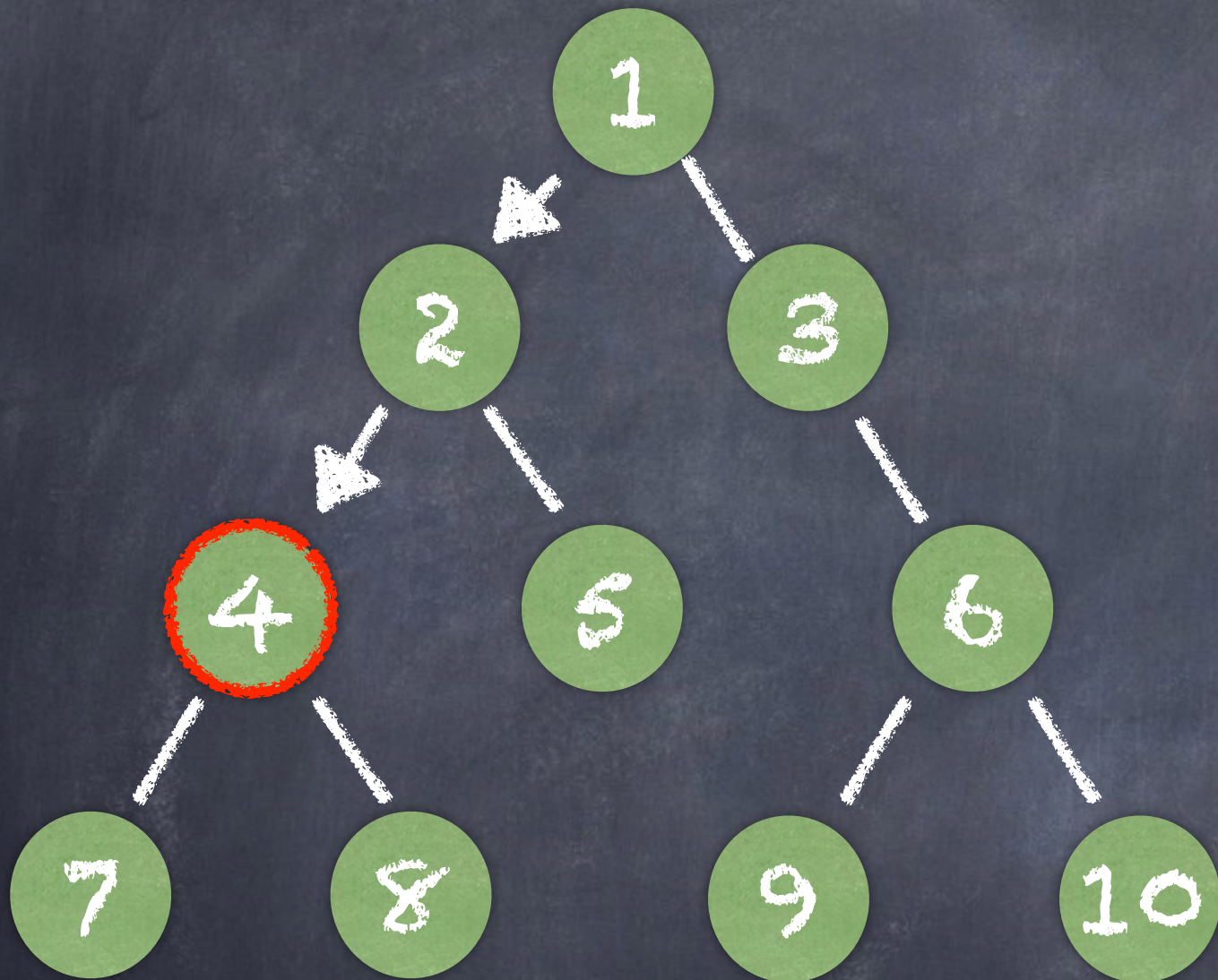
4

2

1

Pizzabestellung

Wiederholung
DFS



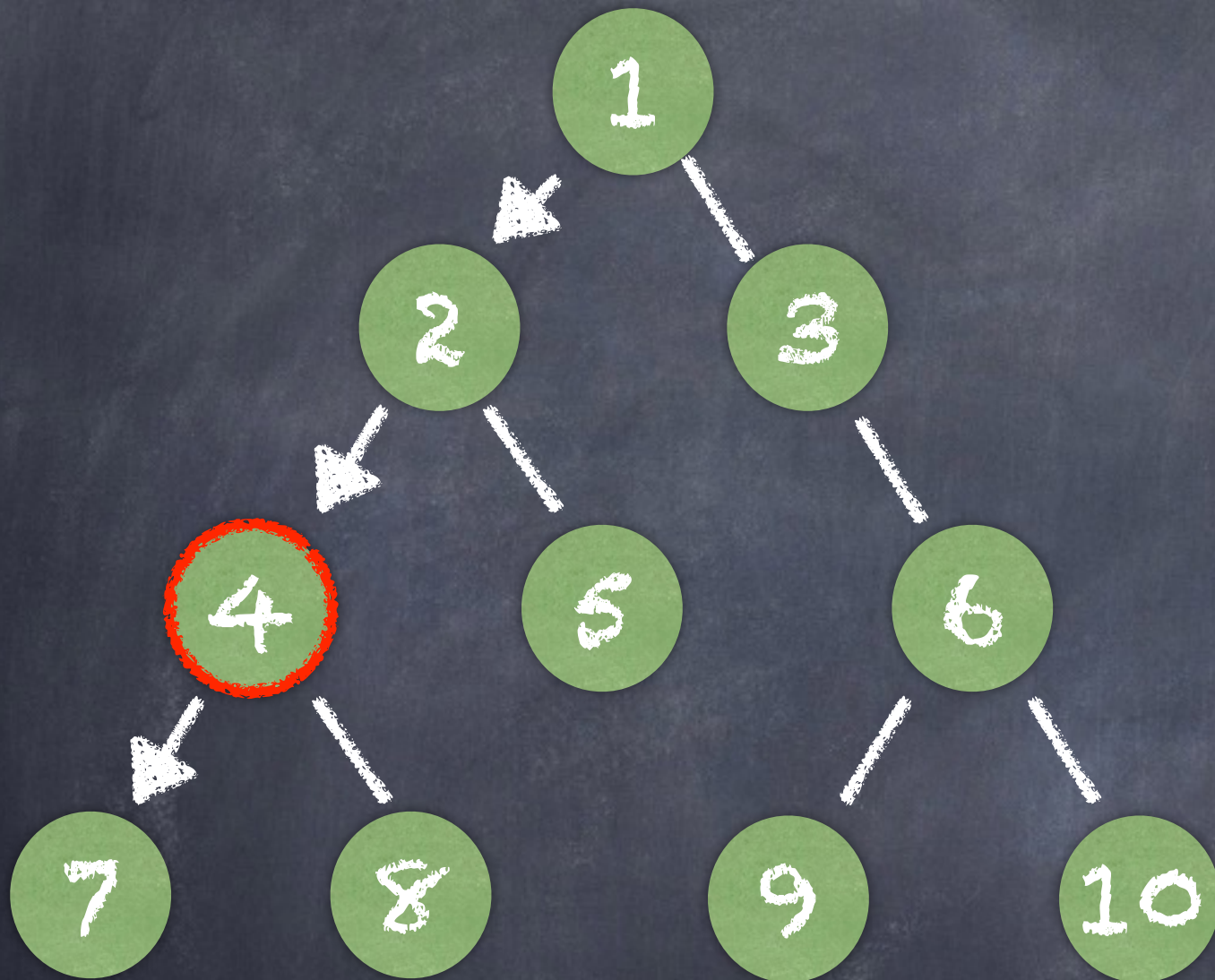
4

2

1

Pizzabestellung

Wiederholung
DFS



7

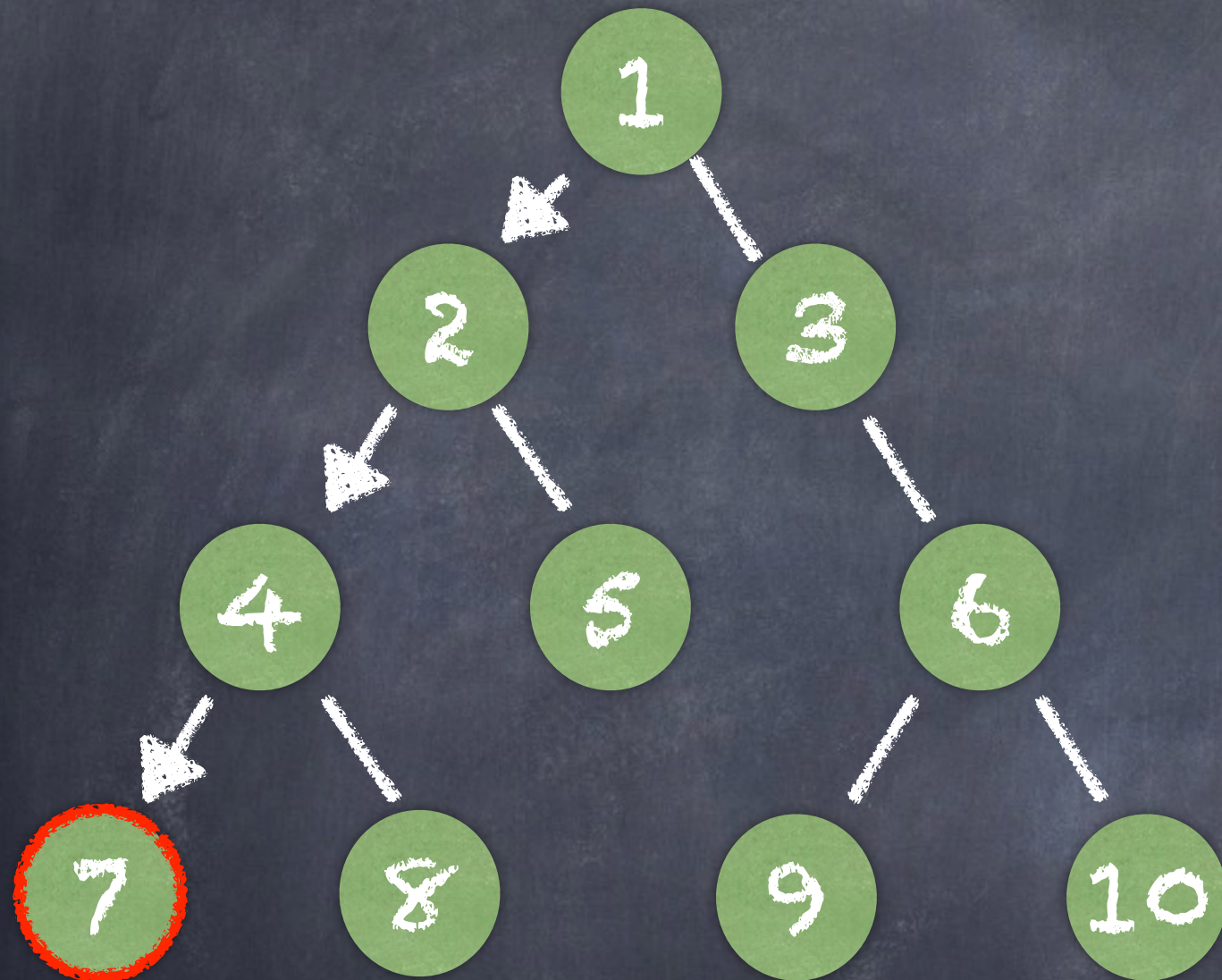
4

2

1

Pizzabestellung

Wiederholung
DFS



7

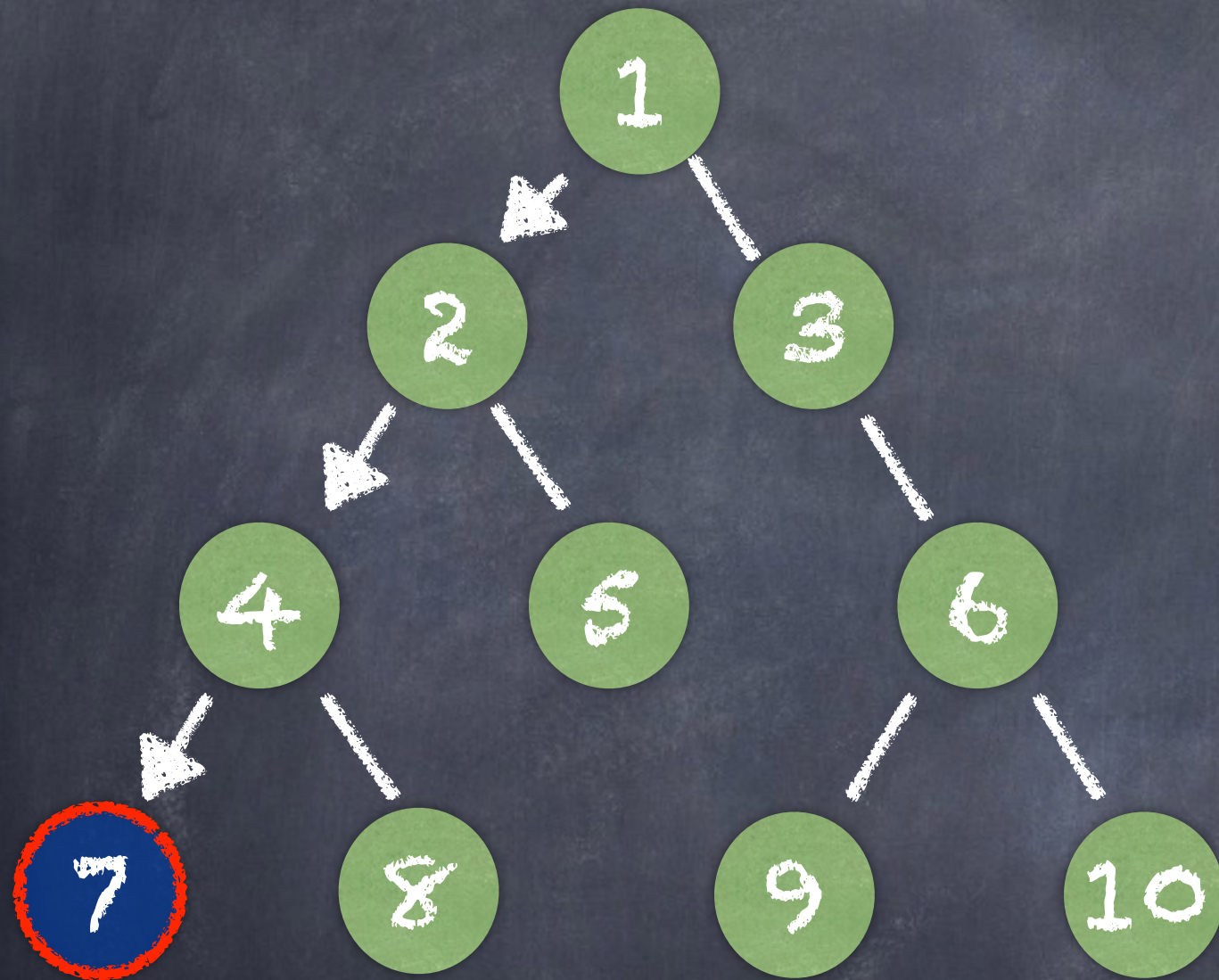
4

2

1

Pizzabestellung

Wiederholung
DFS



7

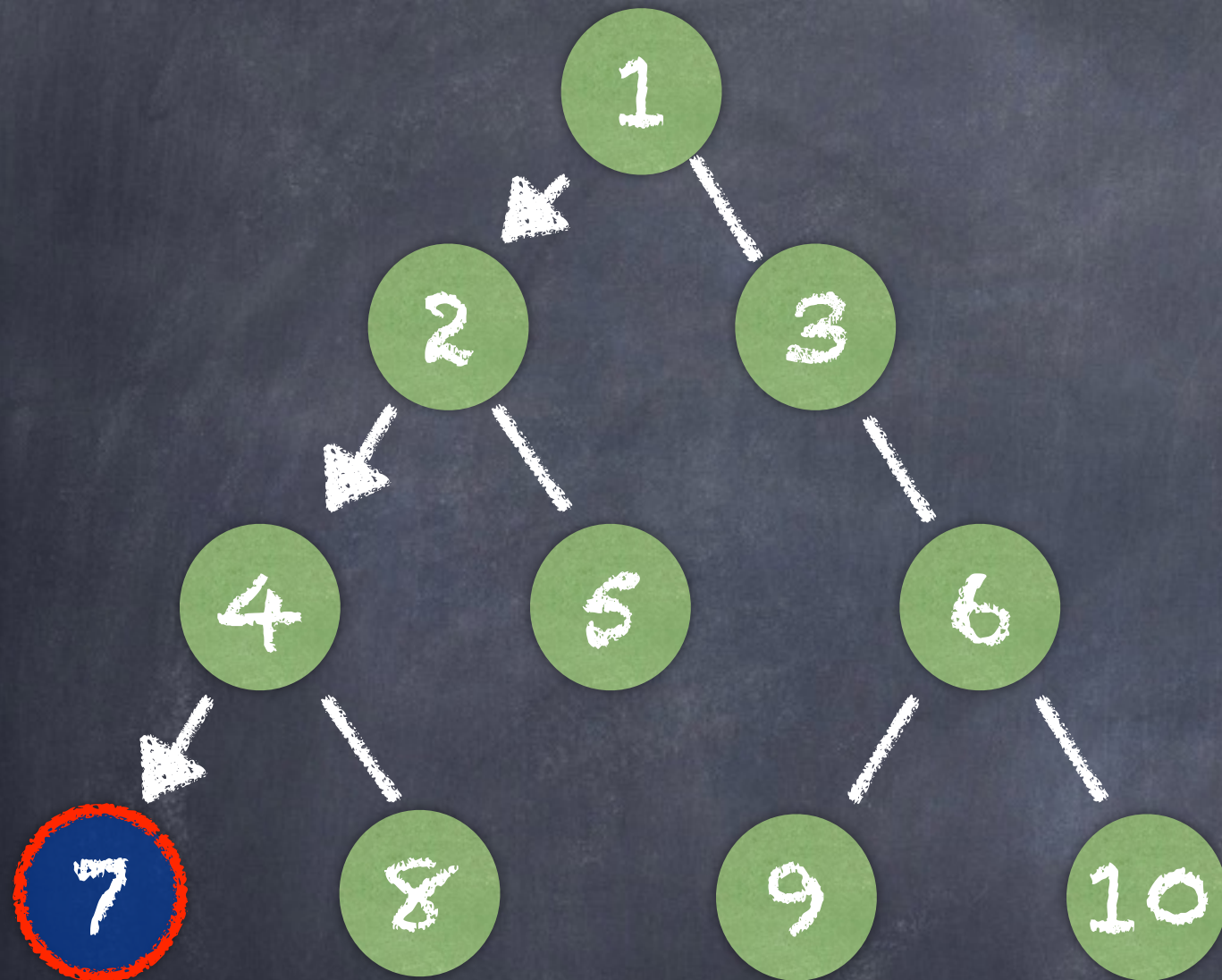
4

2

1

Pizzabestellung

Wiederholung
DFS



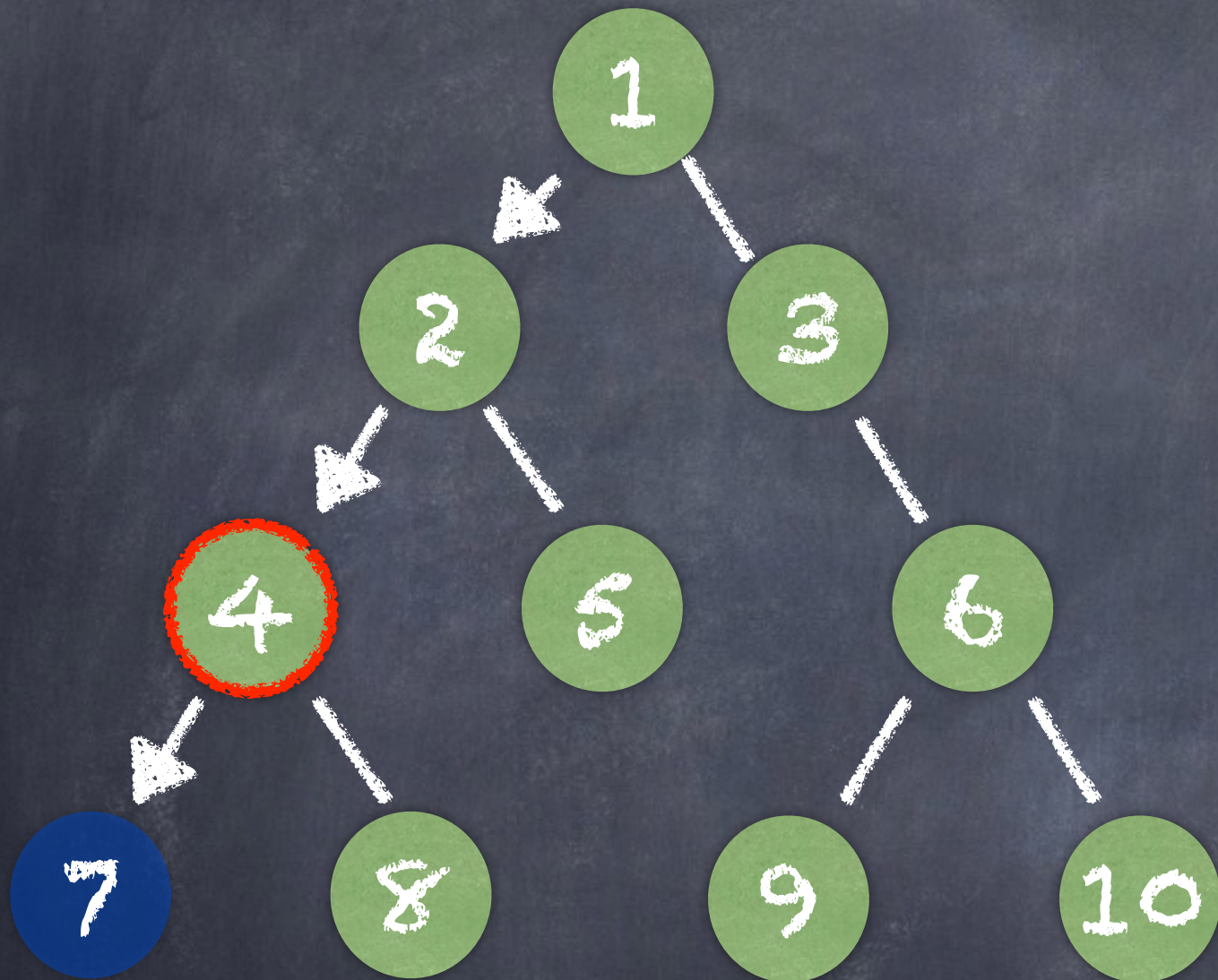
4

2

1

Pizzabestellung

Wiederholung
DFS



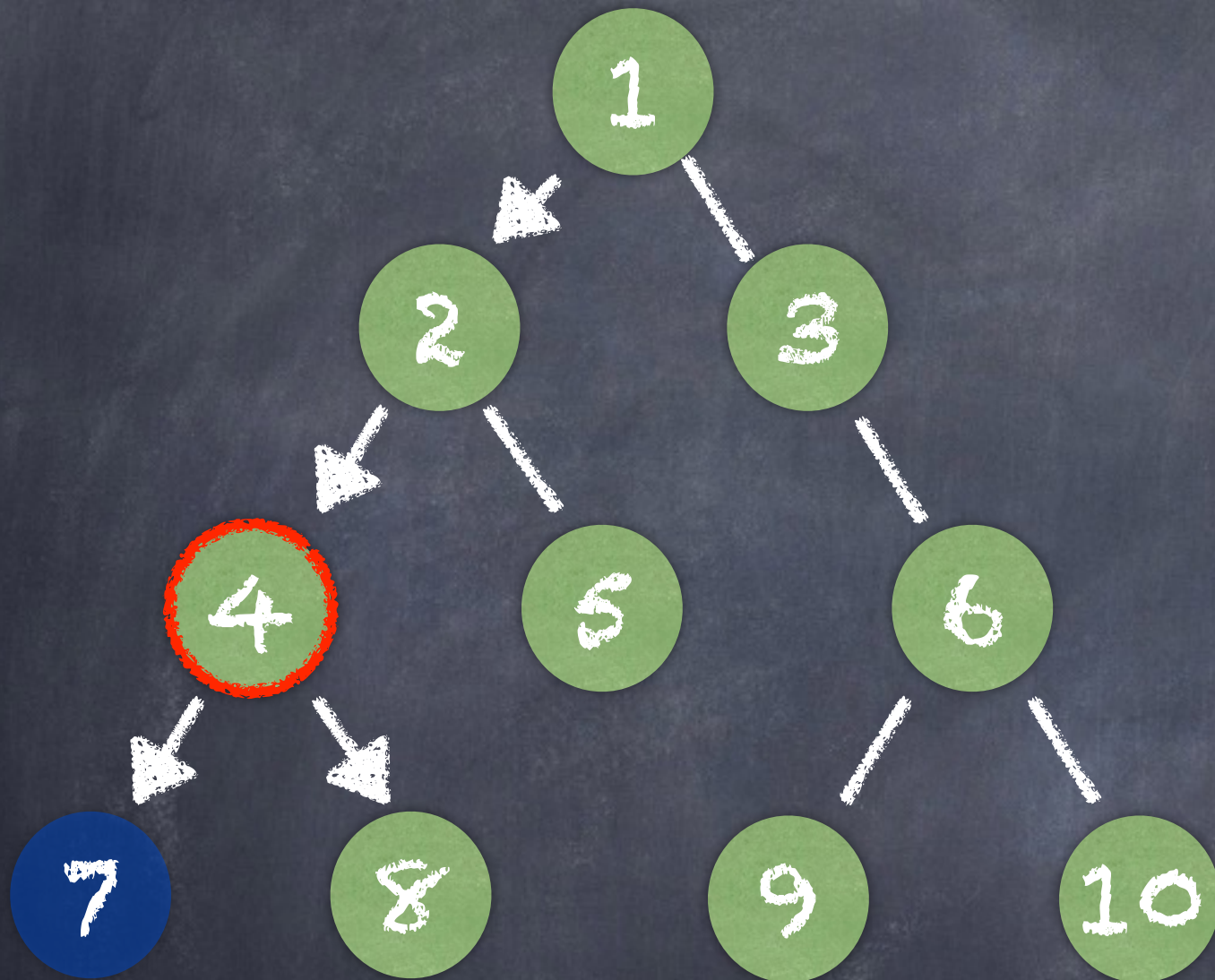
4

2

1

Pizzabestellung

Wiederholung
DFS



8

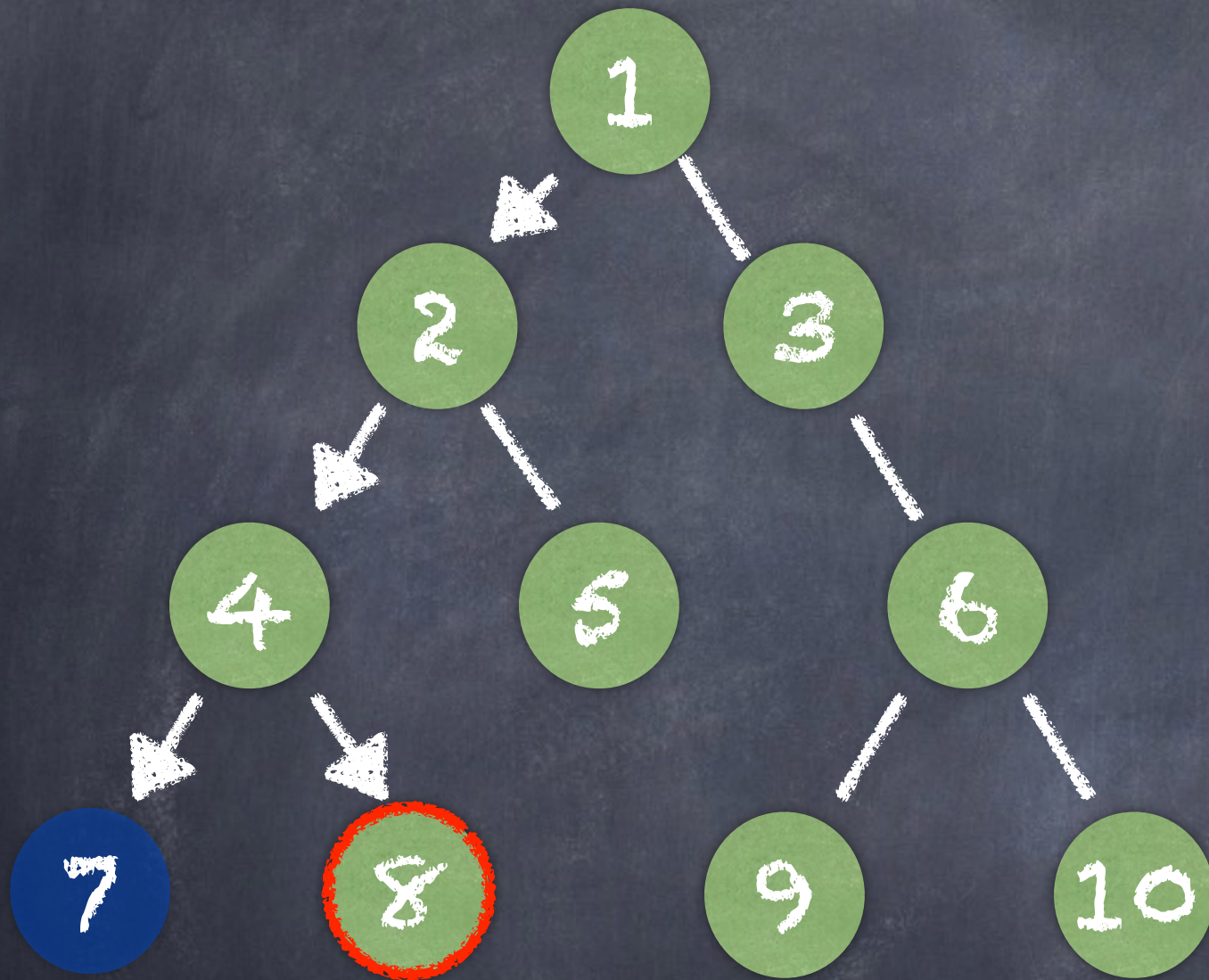
4

2

1

Pizzabestellung

Wiederholung
DFS



8

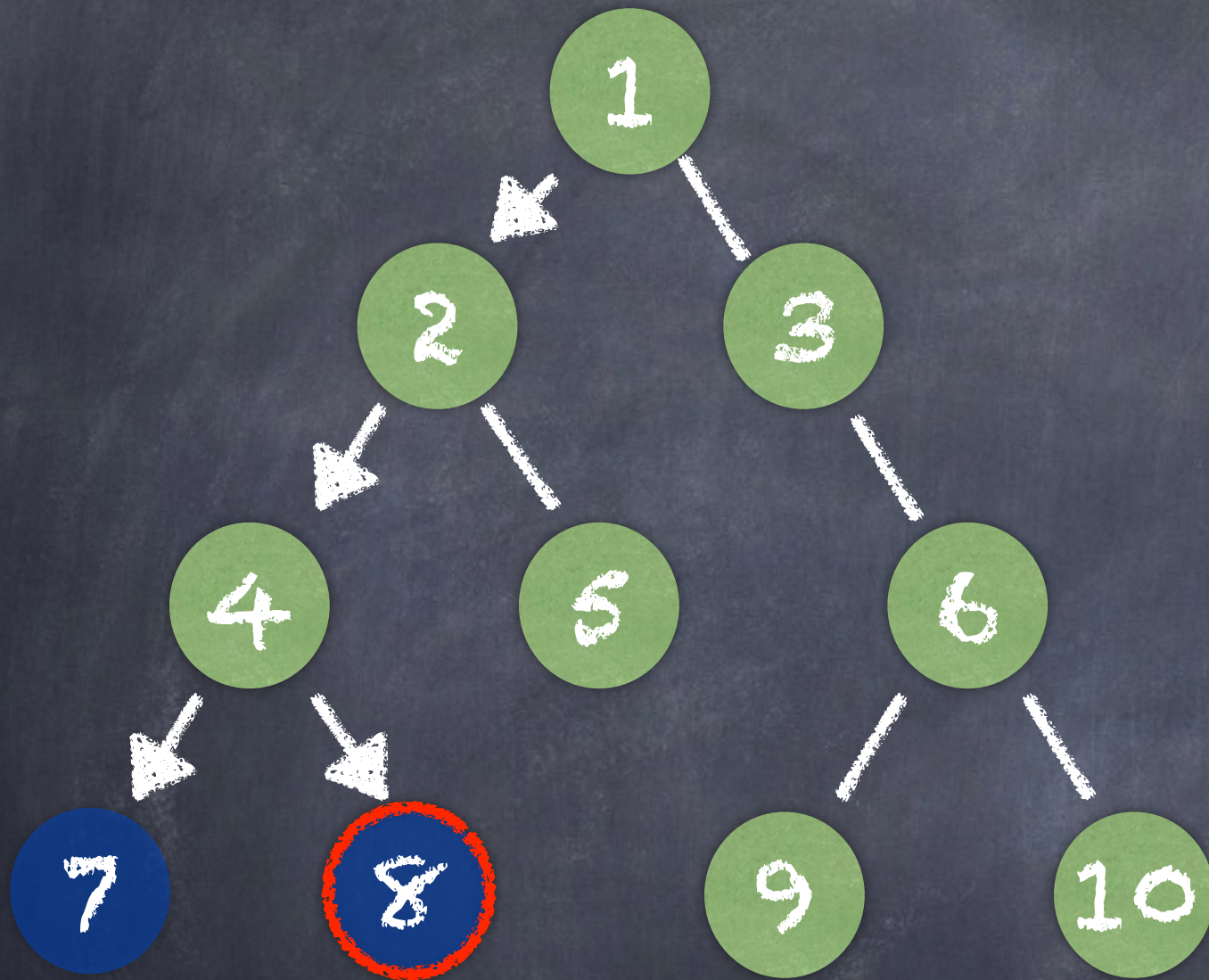
4

2

1

Pizzabestellung

Wiederholung
DFS



8

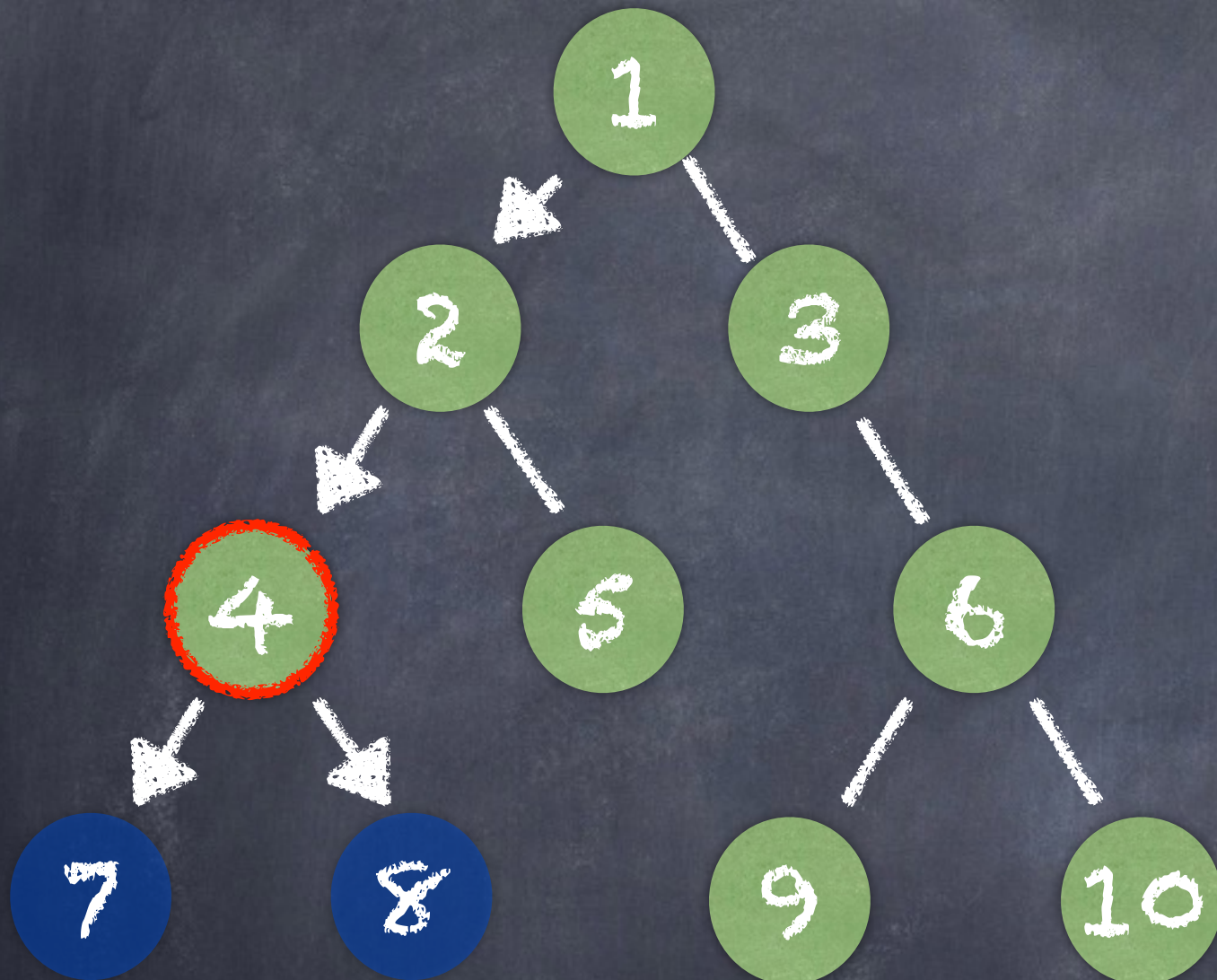
4

2

1

Pizzabestellung

Wiederholung
DFS



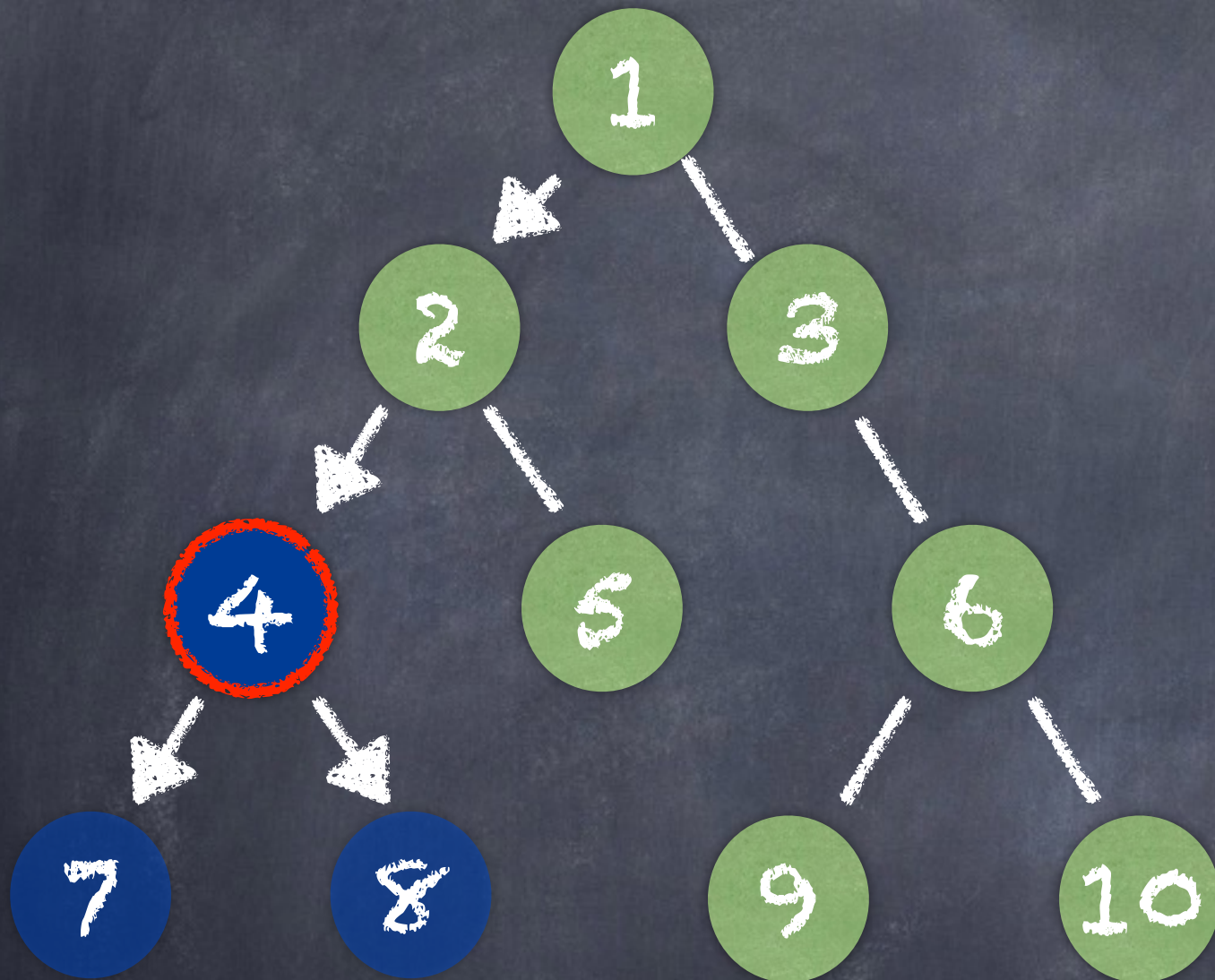
4

2

1

Pizzabestellung

Wiederholung
DFS

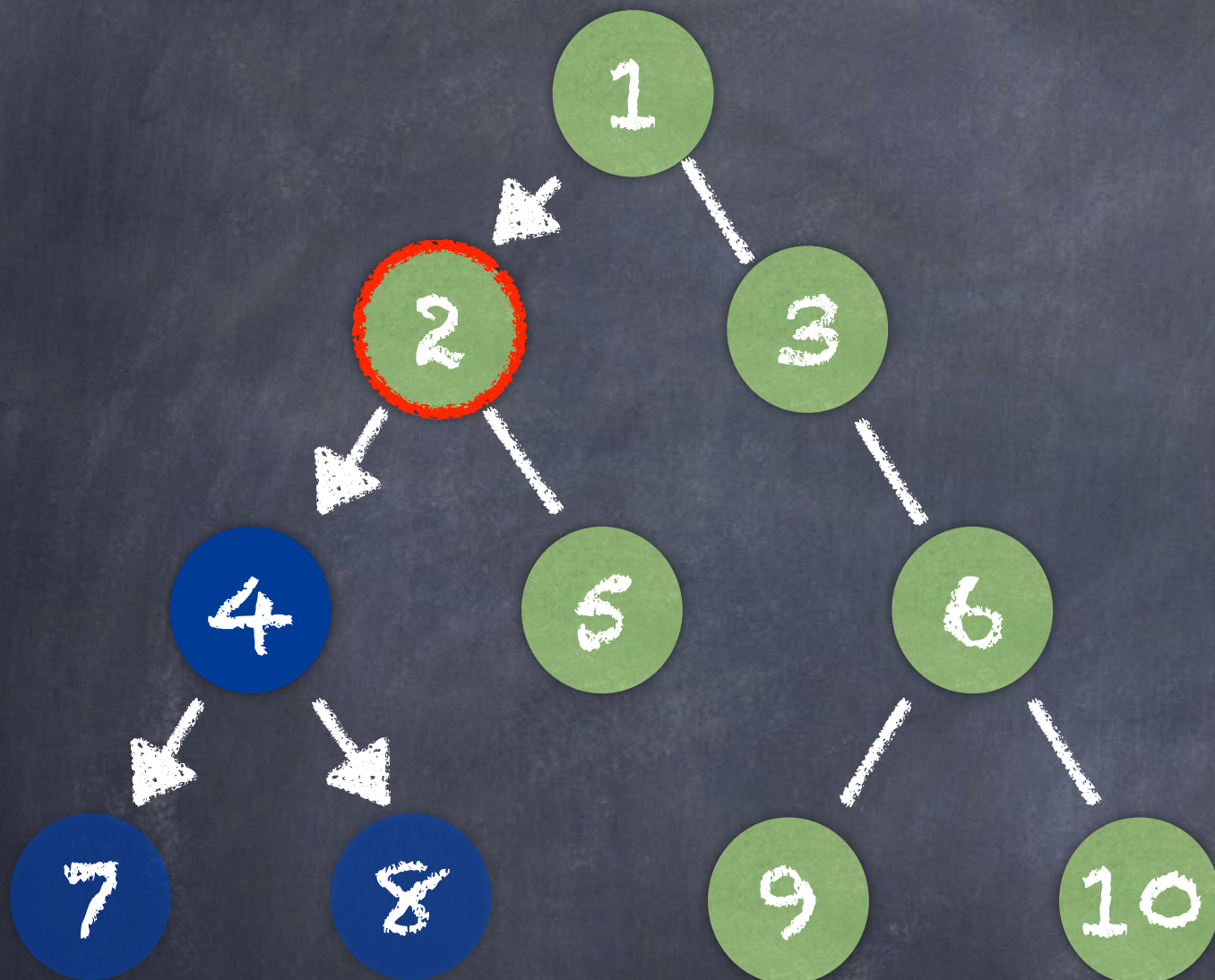


2

1

Pizzabestellung

Wiederholung
DFS

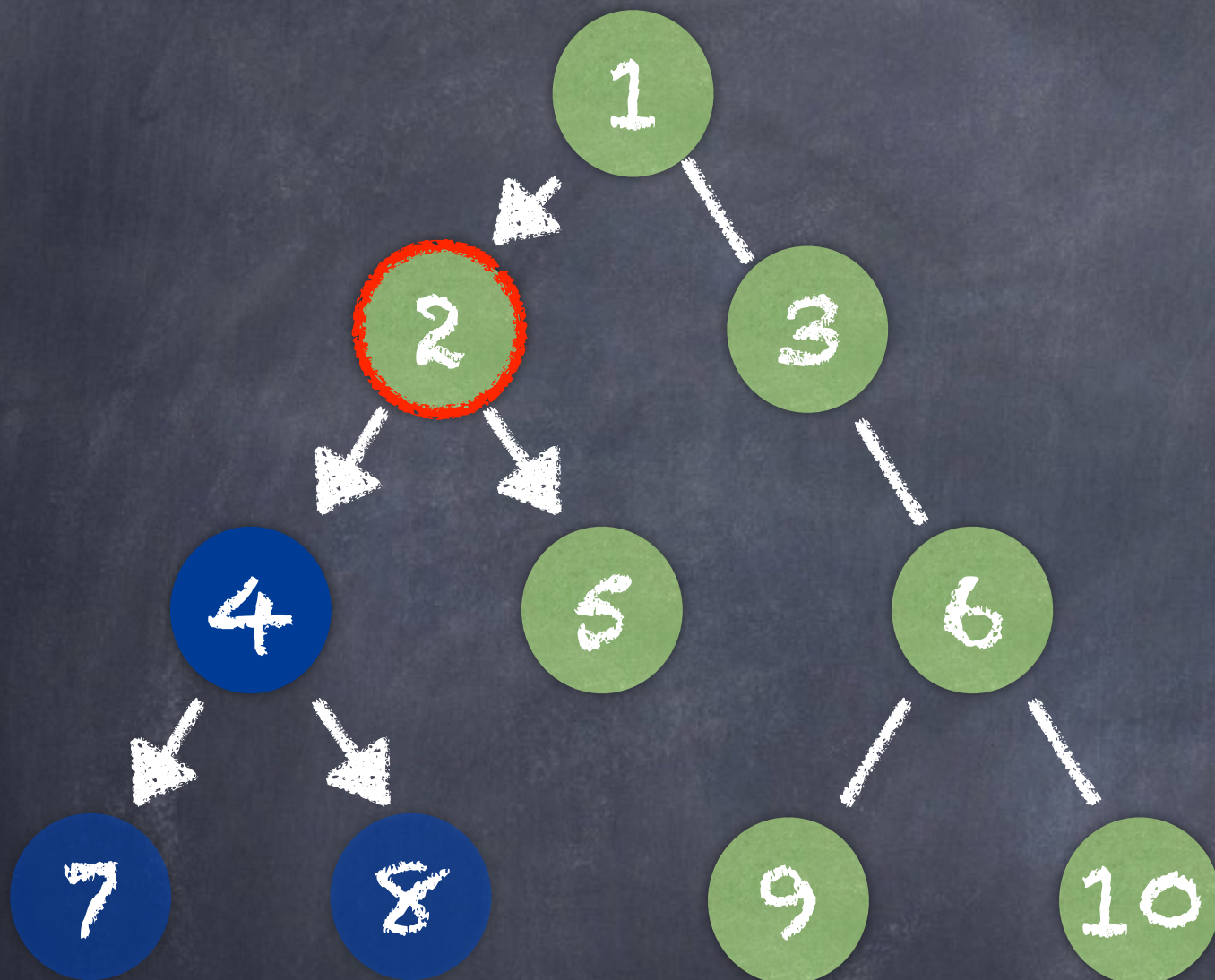


2

1

Pizzabestellung

Wiederholung
DFS



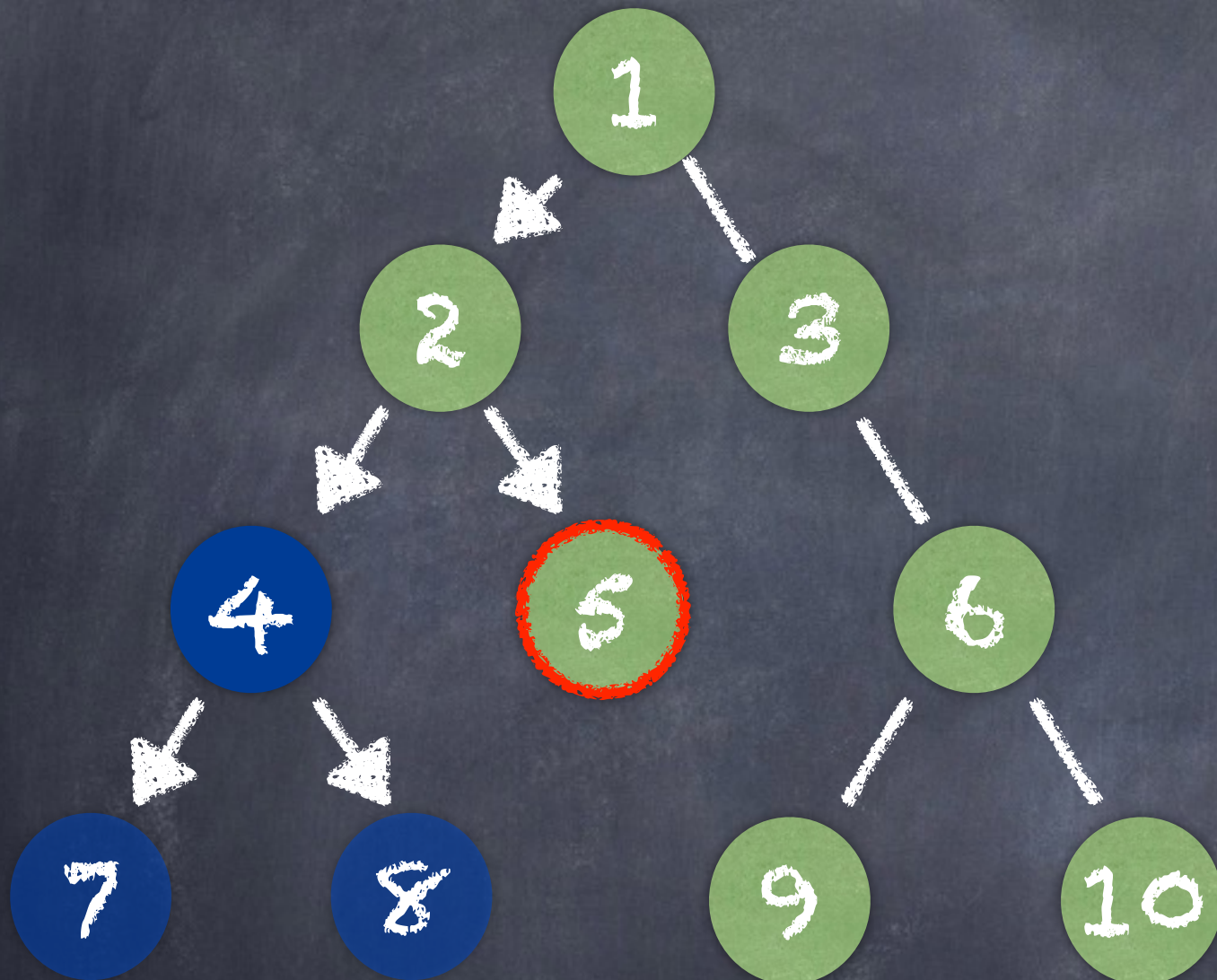
5

2

1

Pizzabestellung

Wiederholung
DFS



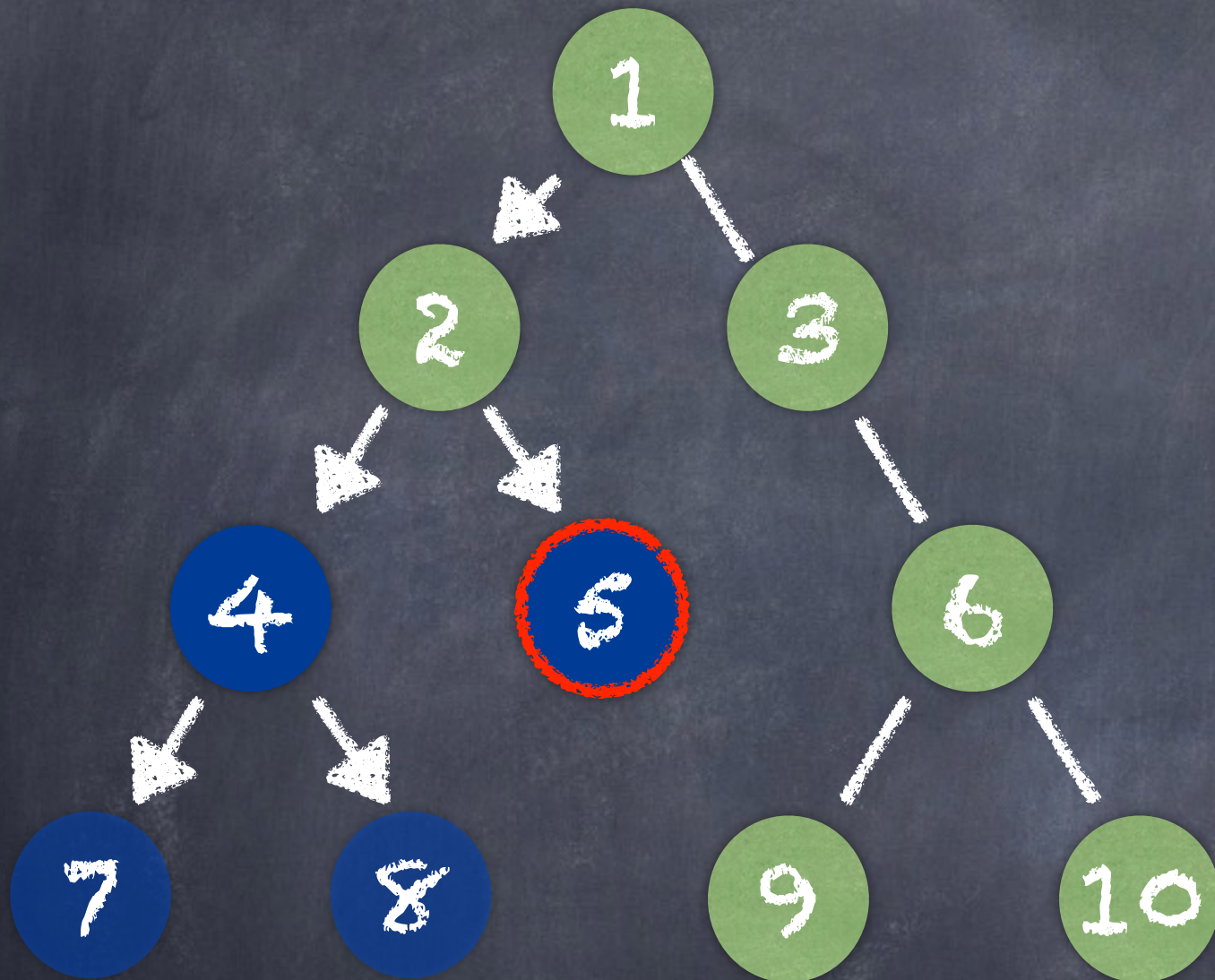
5

2

1

Pizzabestellung

Wiederholung
DFS

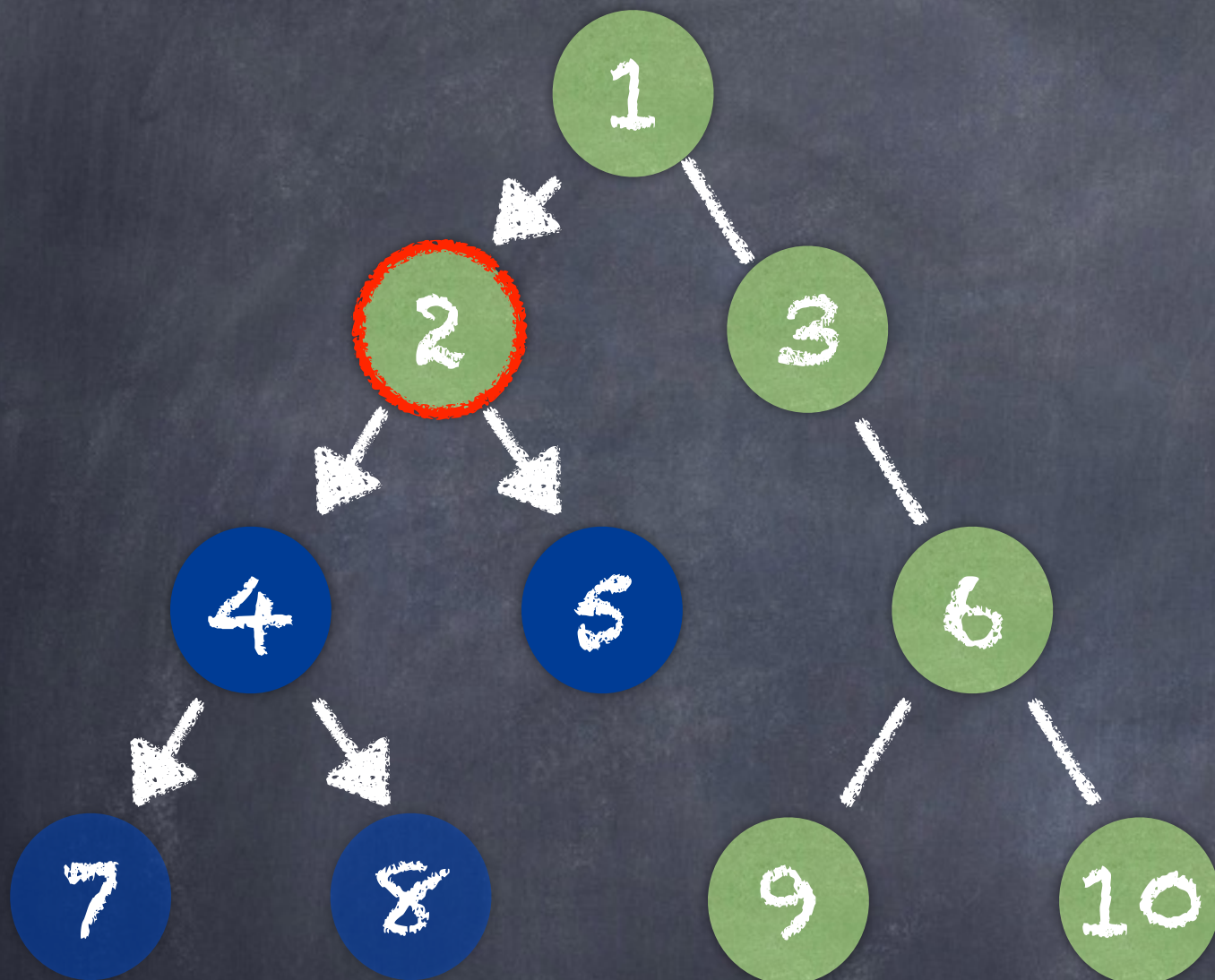


2

1

Pizzabestellung

Wiederholung
DFS

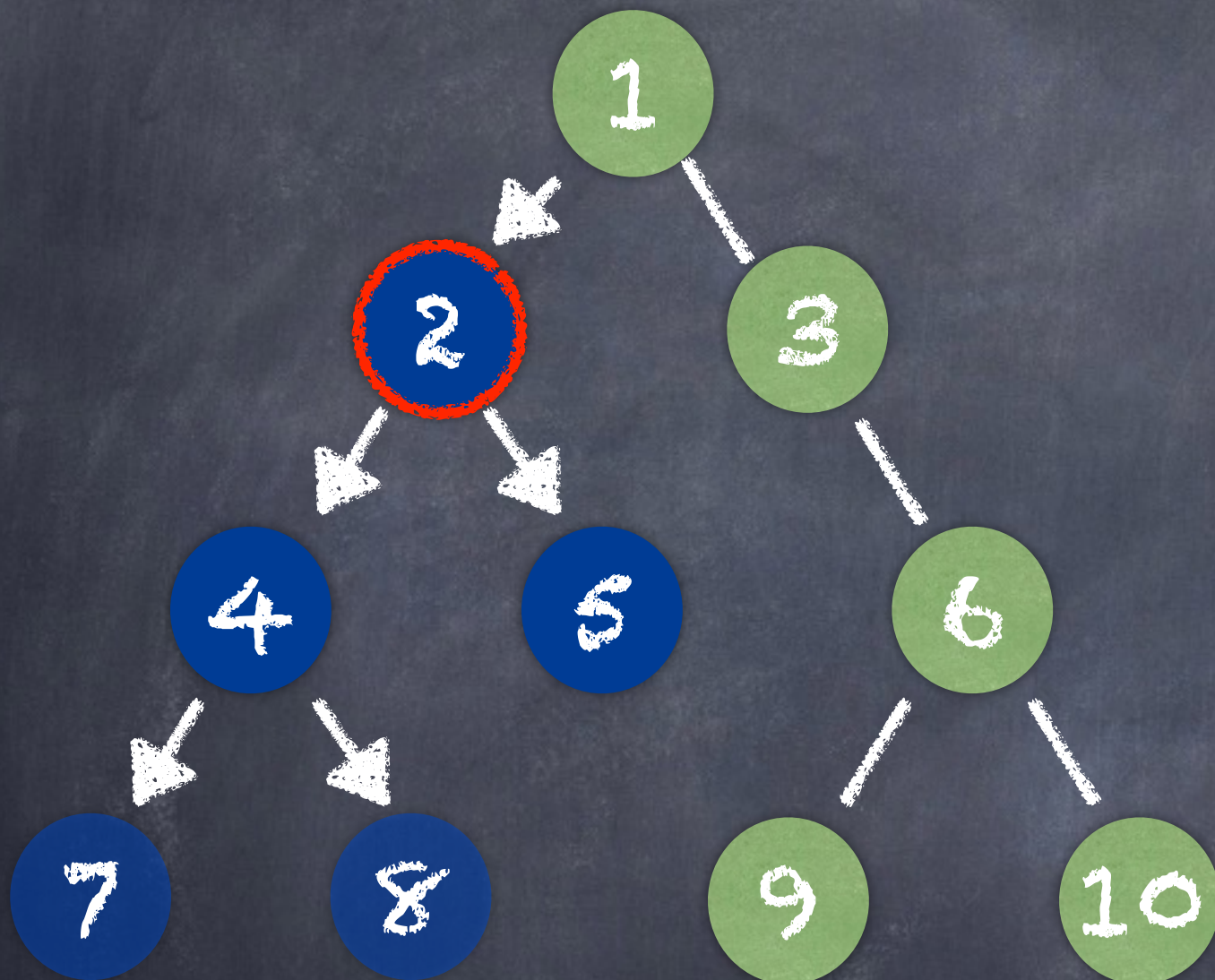


2

1

Pizzabestellung

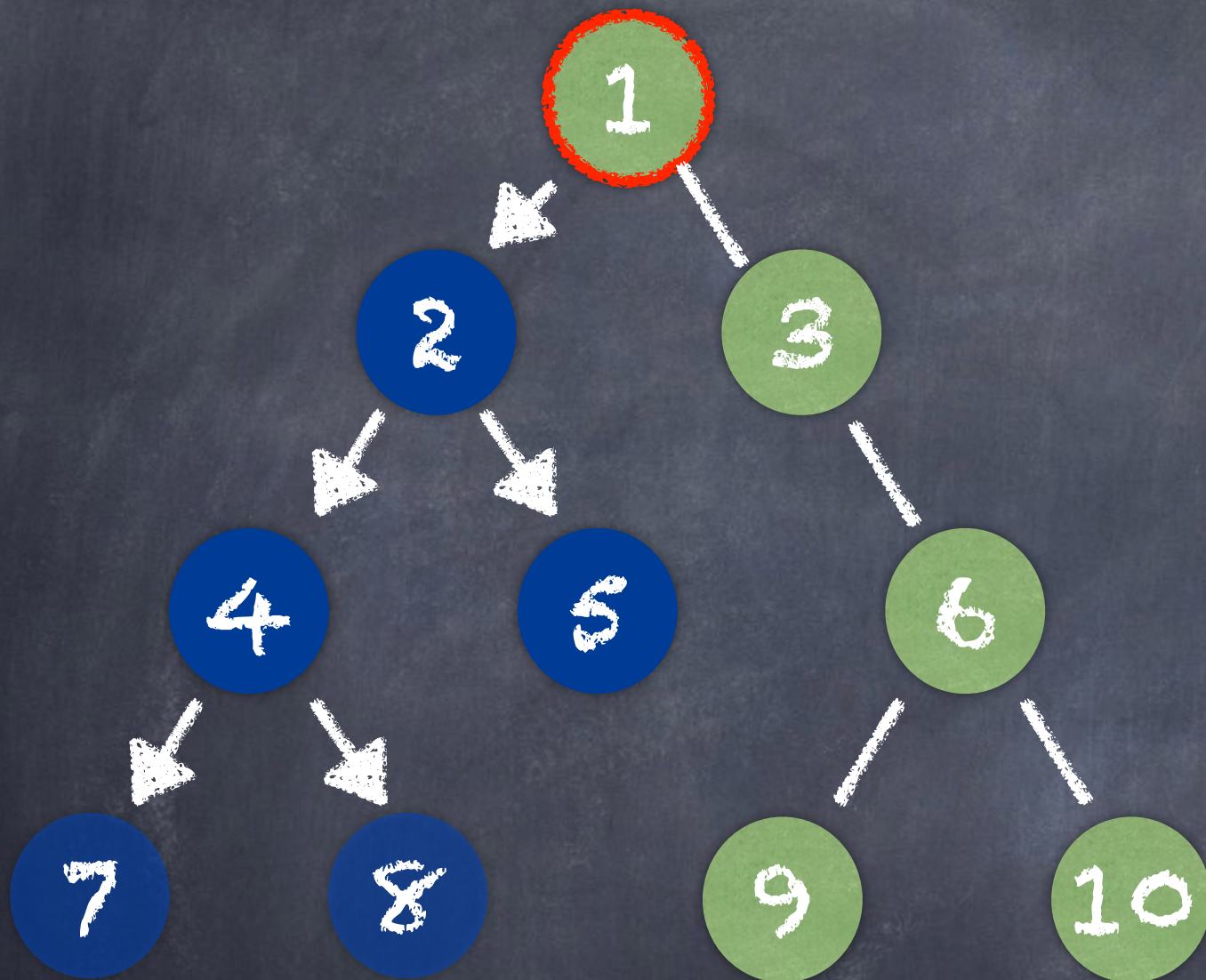
Wiederholung
DFS



1

Pizzabestellung

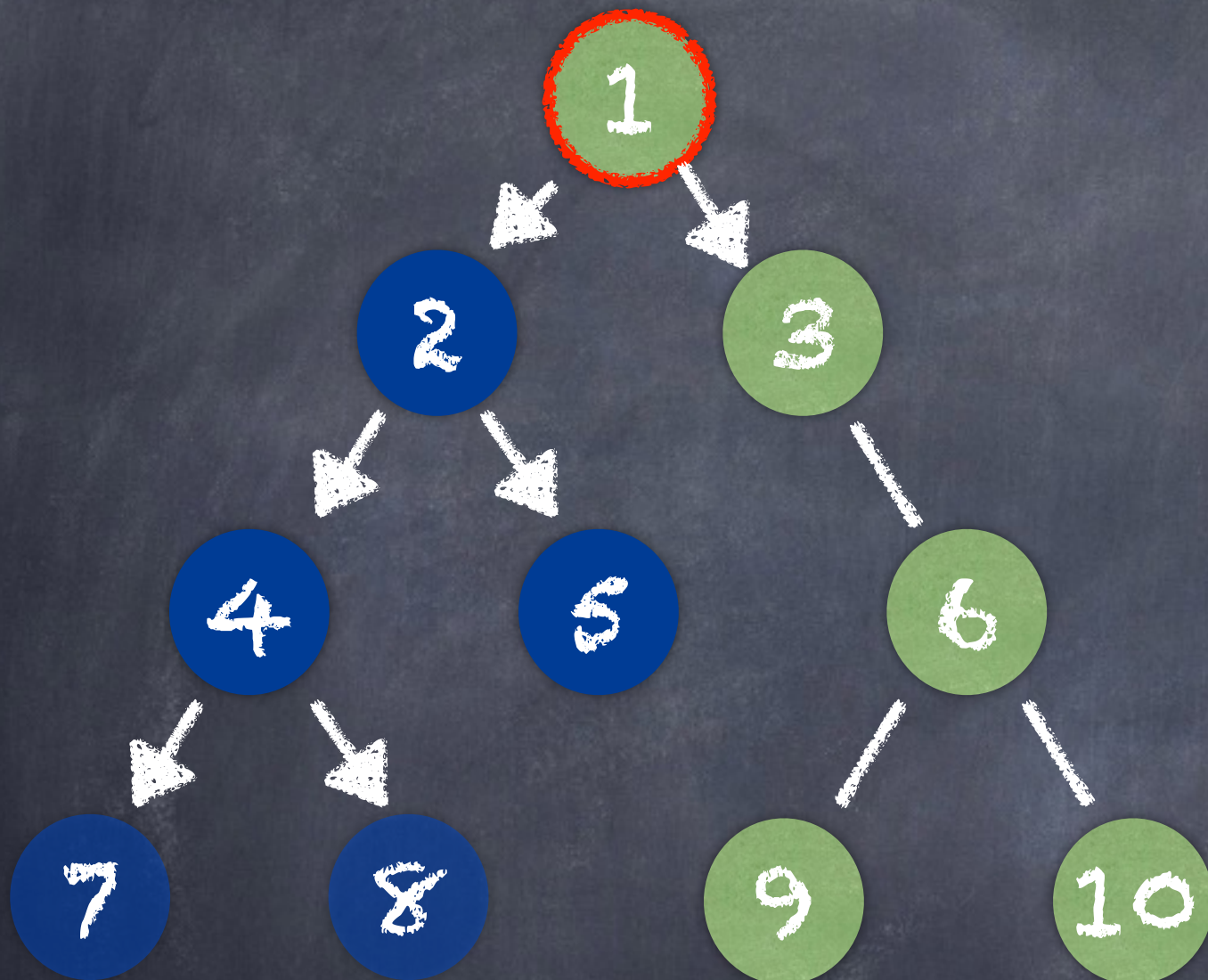
Wiederholung
DFS



1

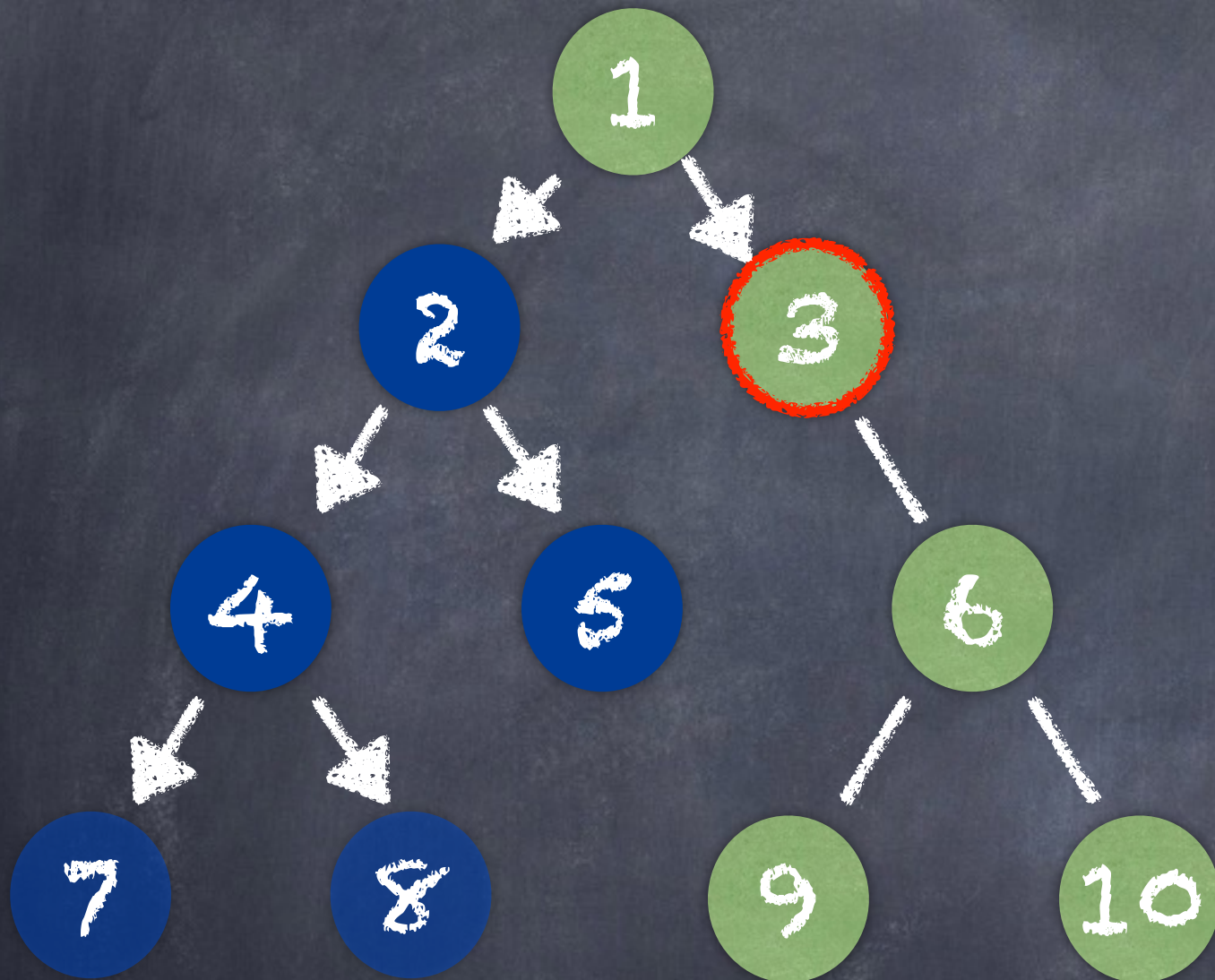
Pizzabestellung

Wiederholung
DFS



Pizzabestellung

Wiederholung
DFS

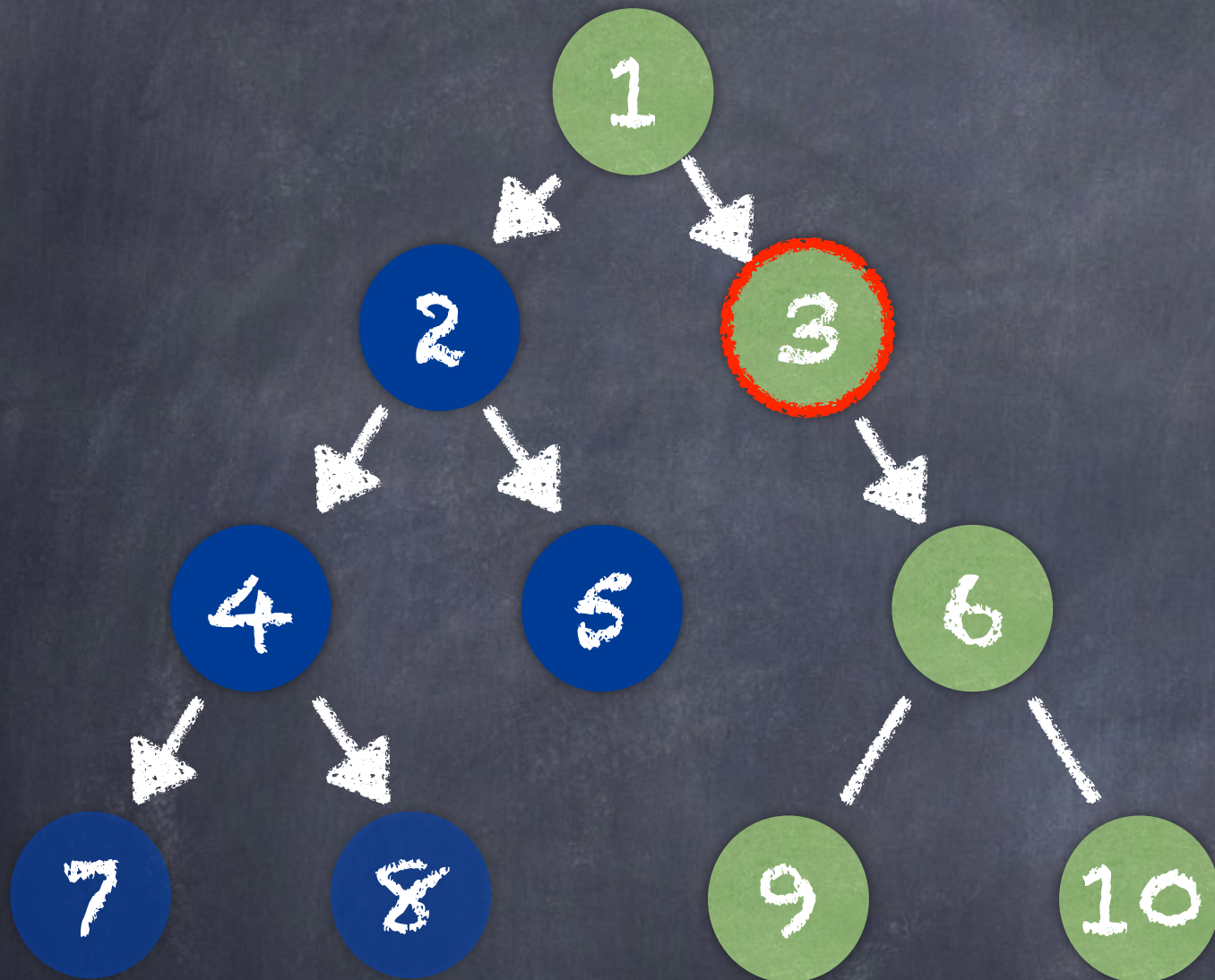


3

1

Pizzabestellung

Wiederholung
DFS



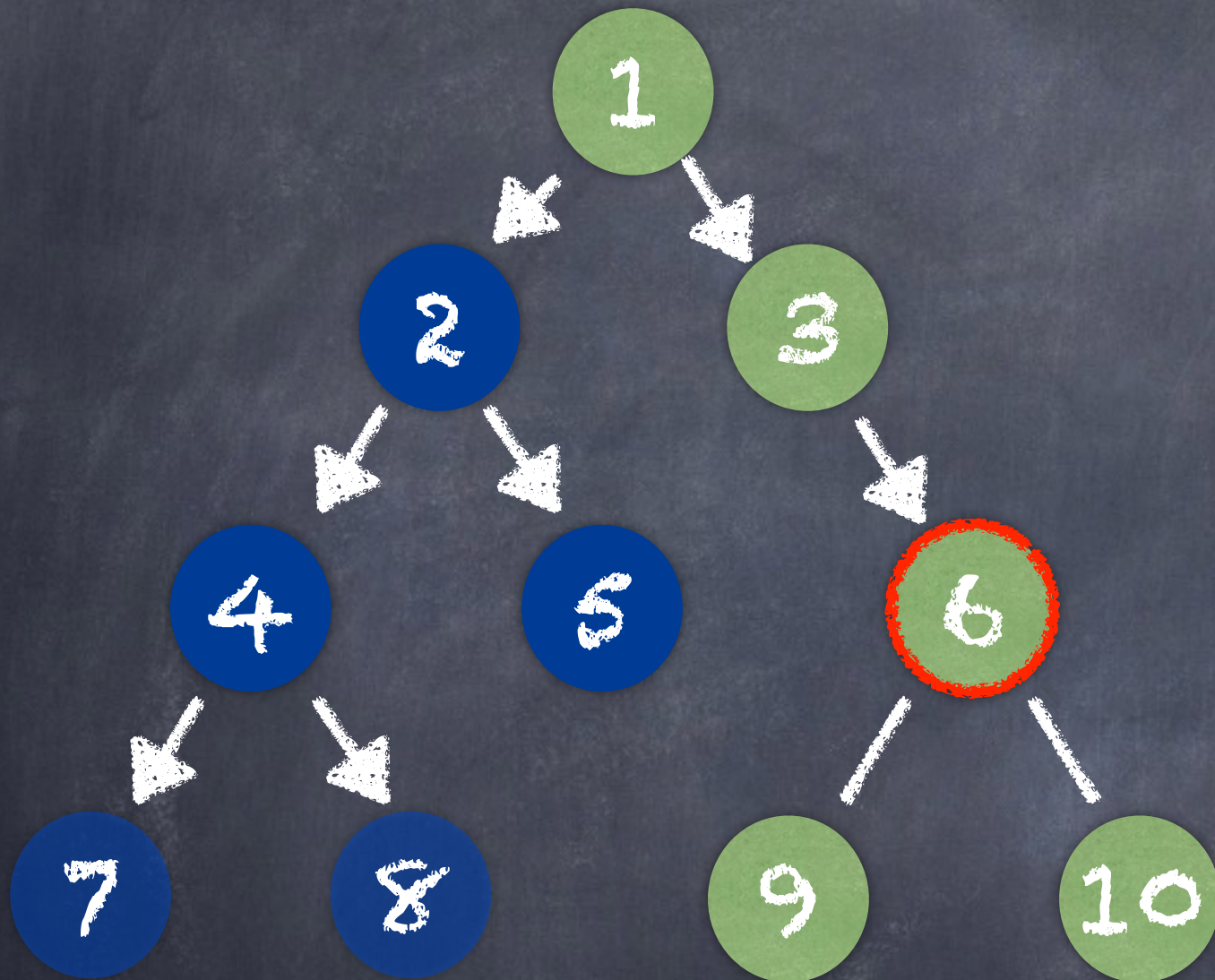
6

3

1

Pizzabestellung

Wiederholung
DFS



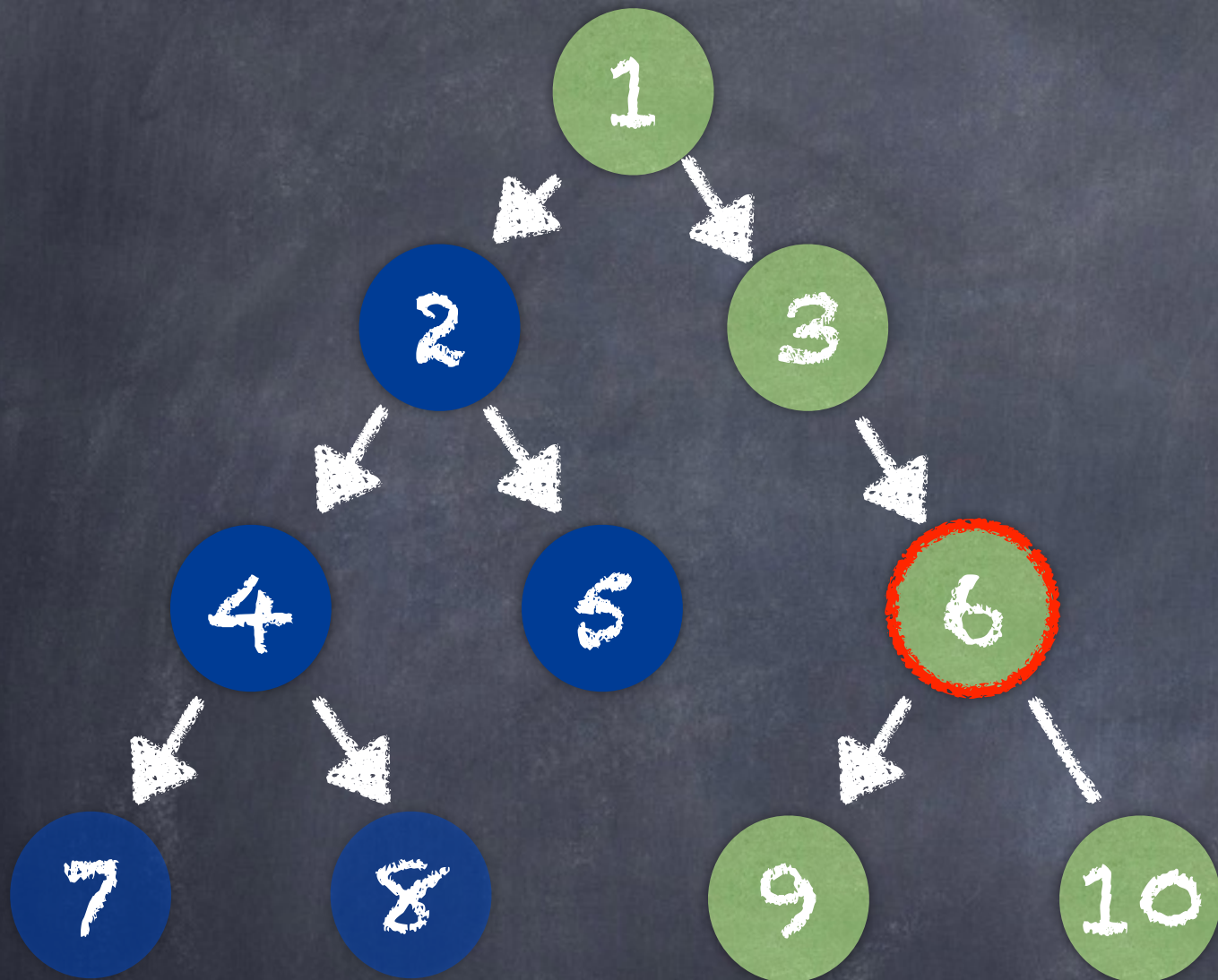
6

3

1

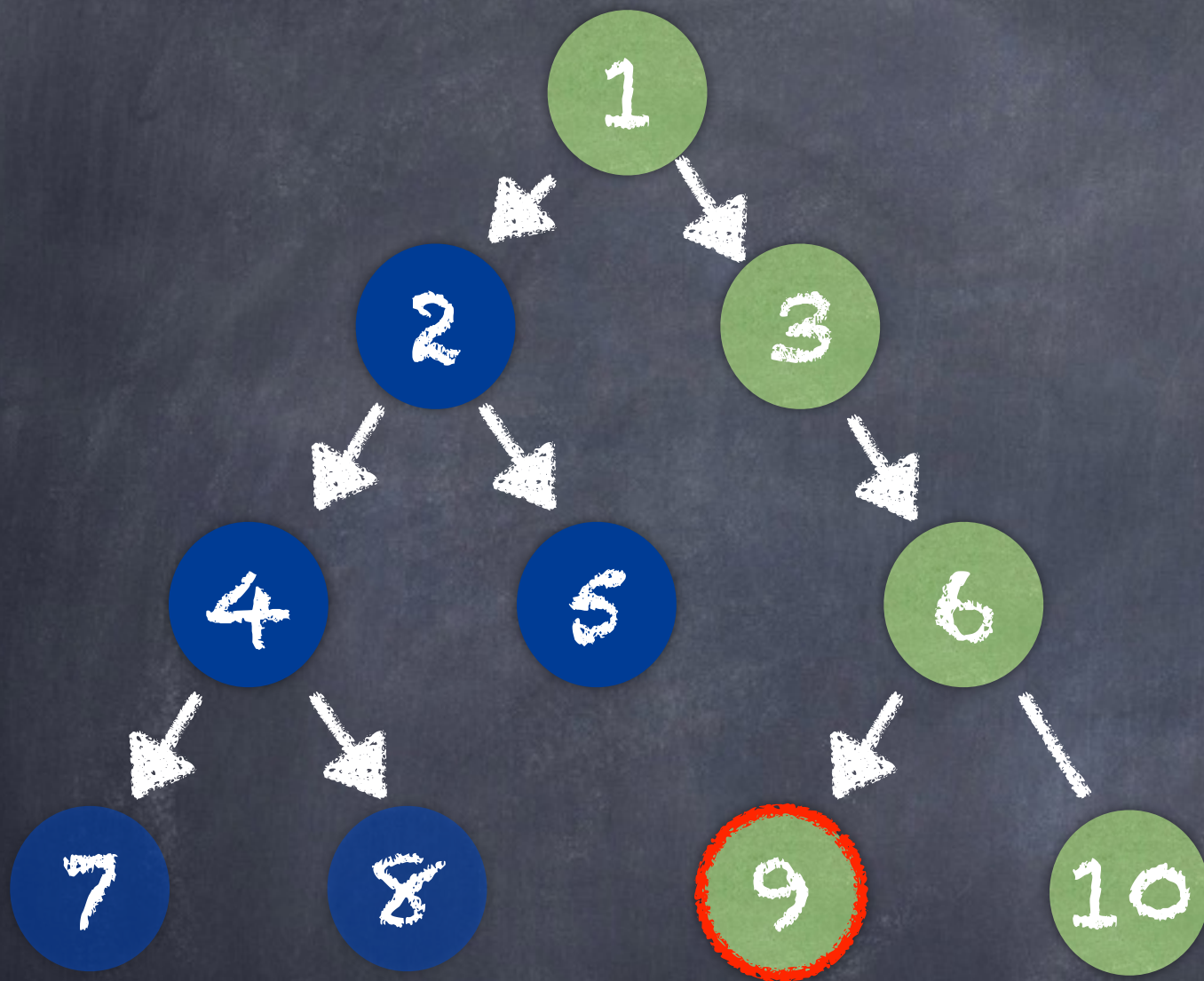
Pizzabestellung

Wiederholung
DFS



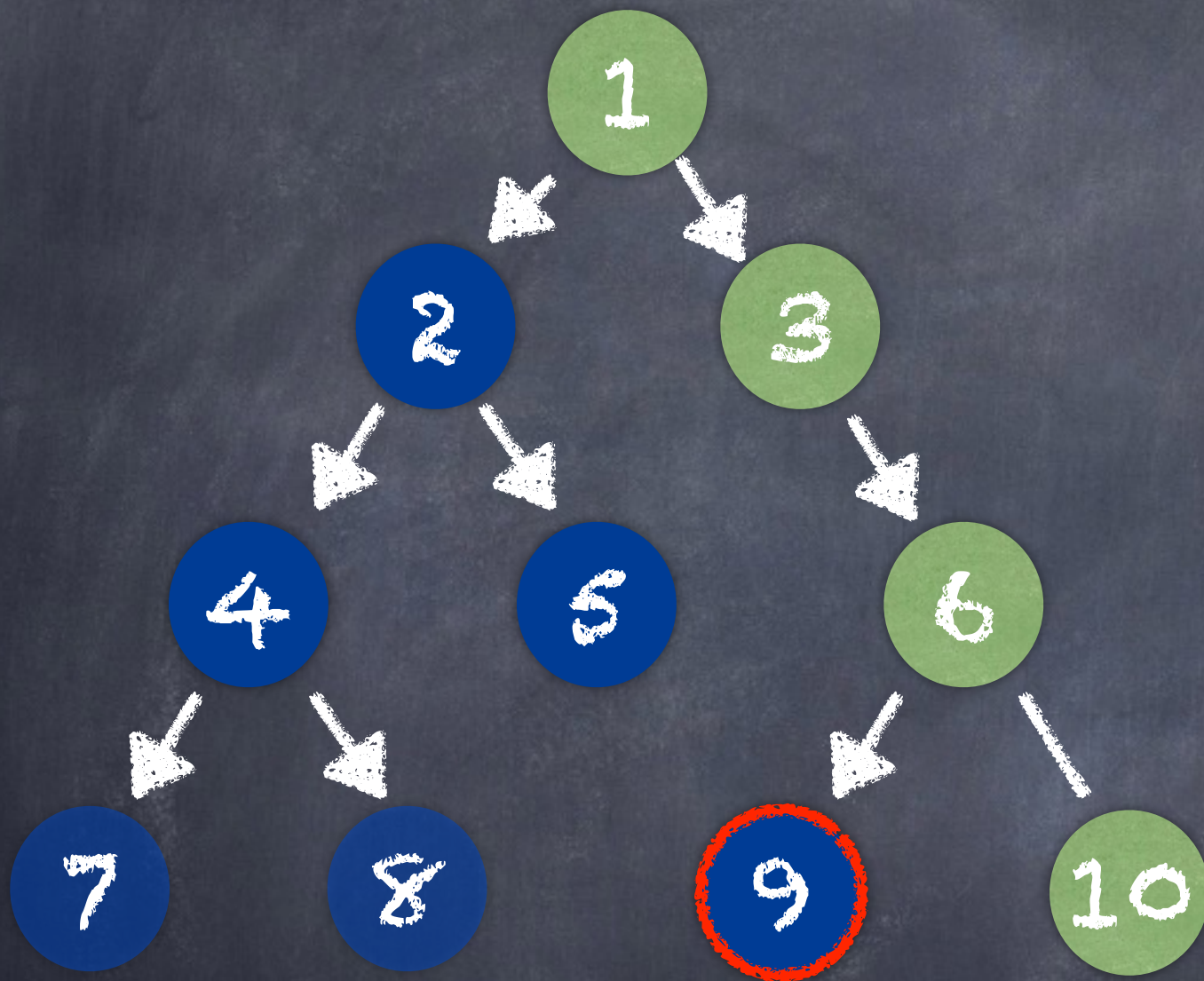
Pizzabestellung

Wiederholung
DFS



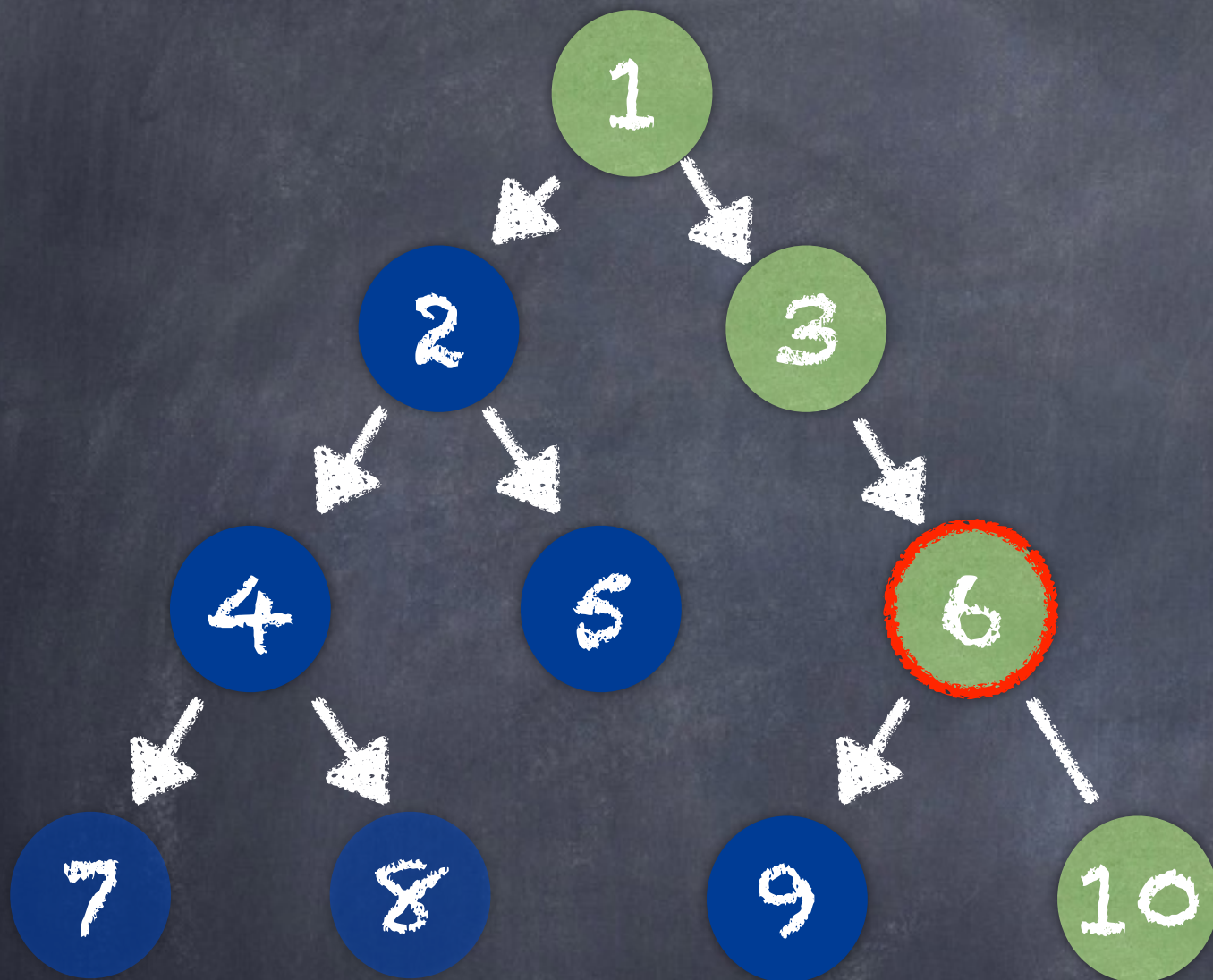
Pizzabestellung

Wiederholung
DFS



Pizzabestellung

Wiederholung
DFS



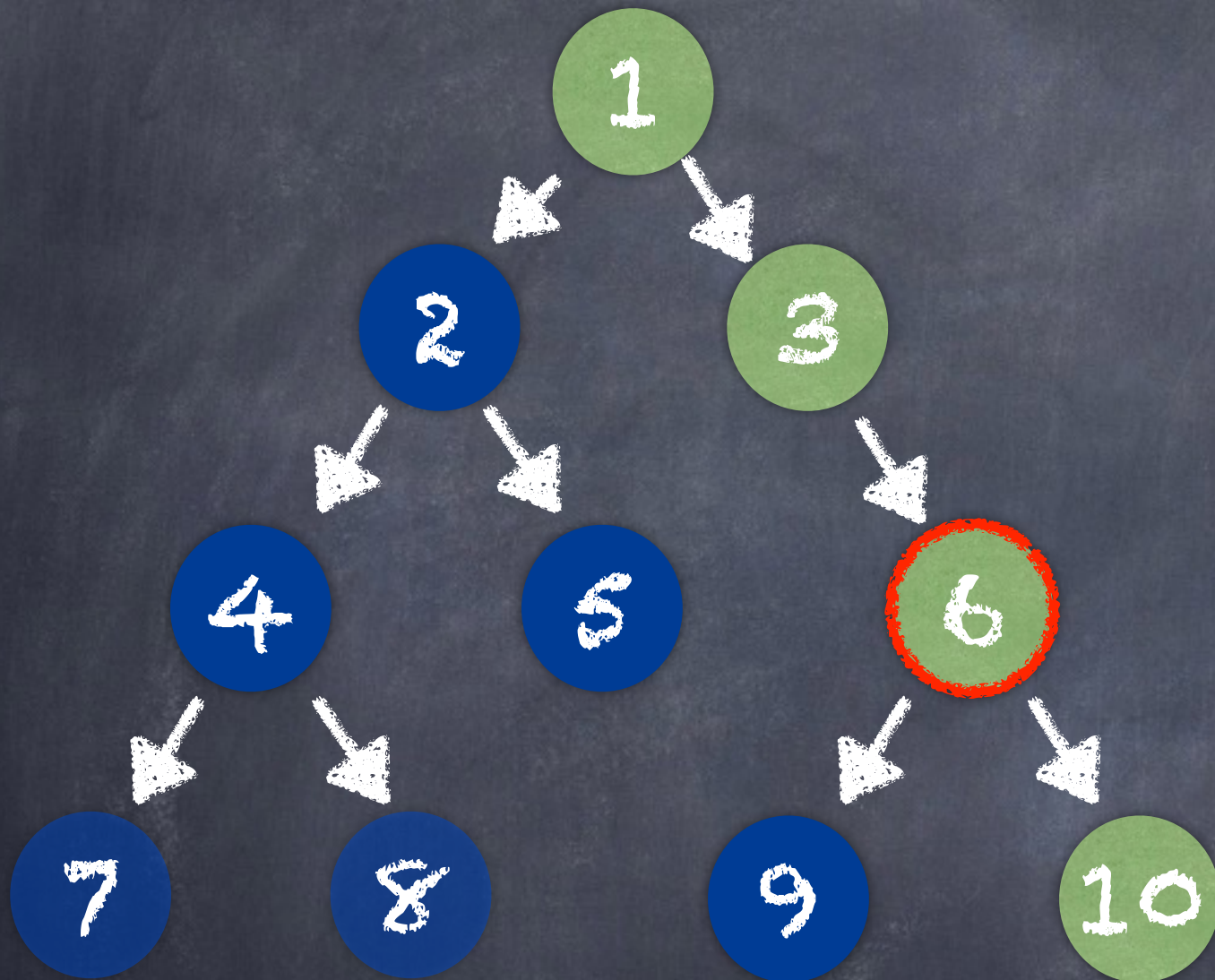
6

3

1

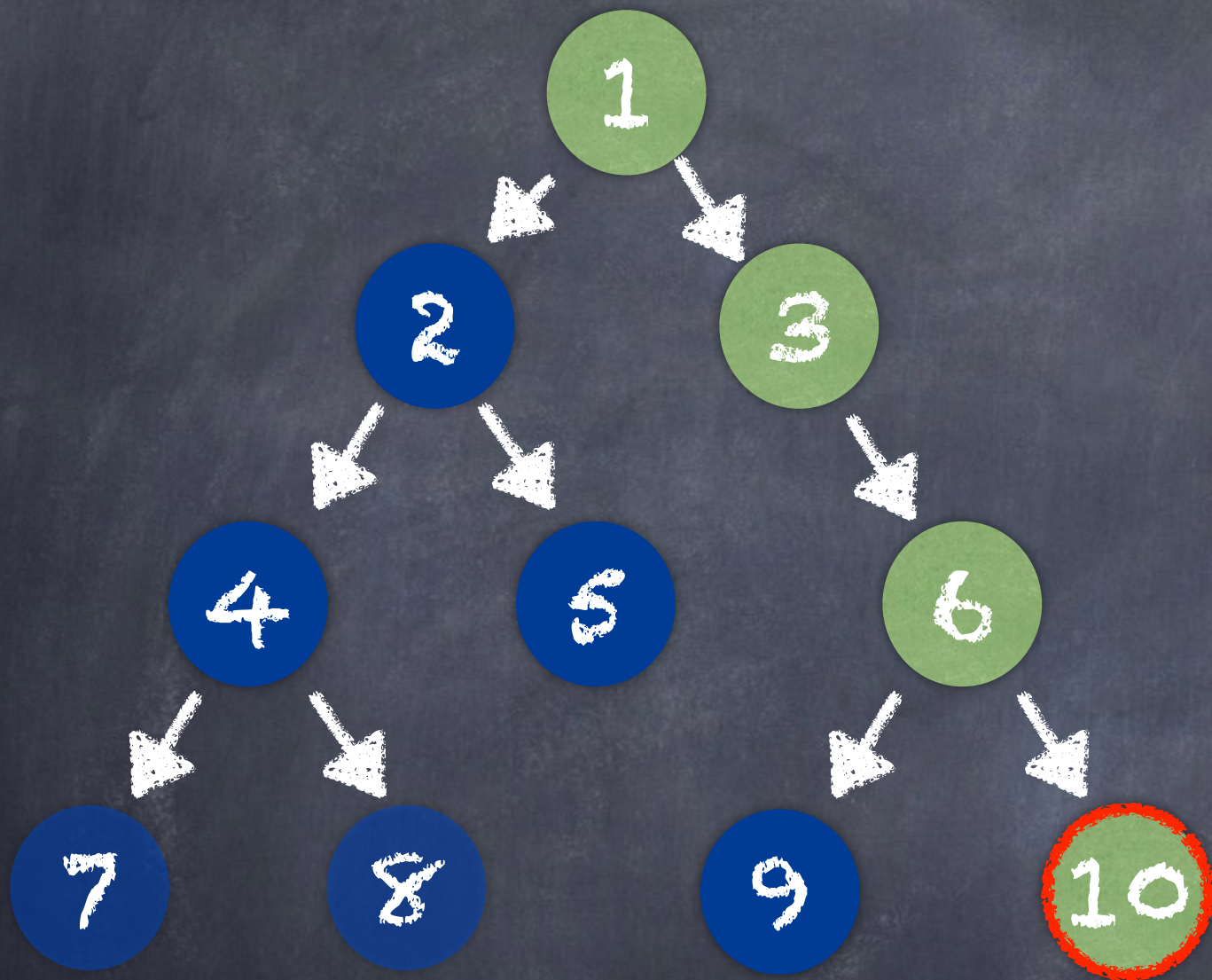
Pizzabestellung

Wiederholung
DFS



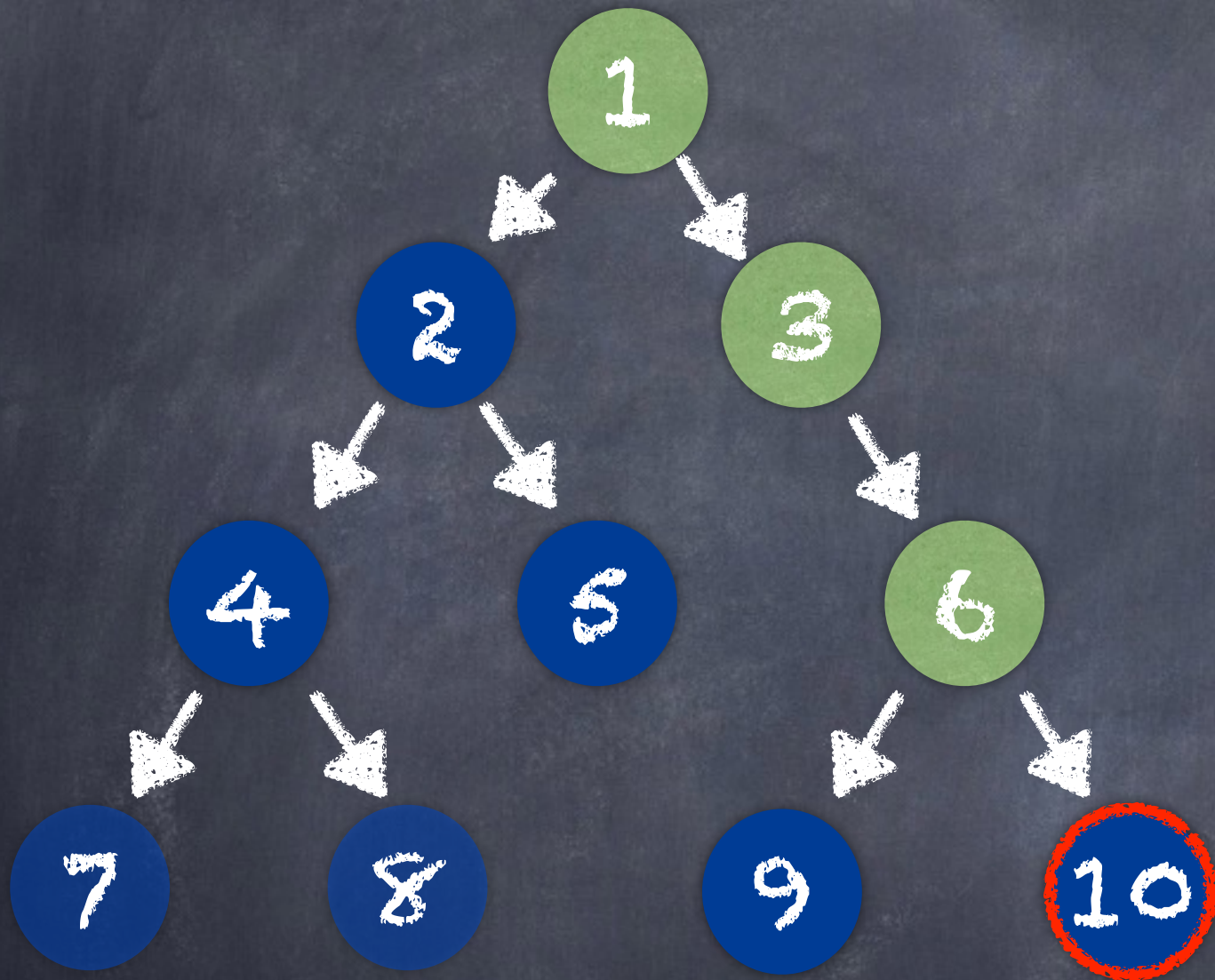
Pizzabestellung

Wiederholung
DFS



Pizzabestellung

Wiederholung
DFS



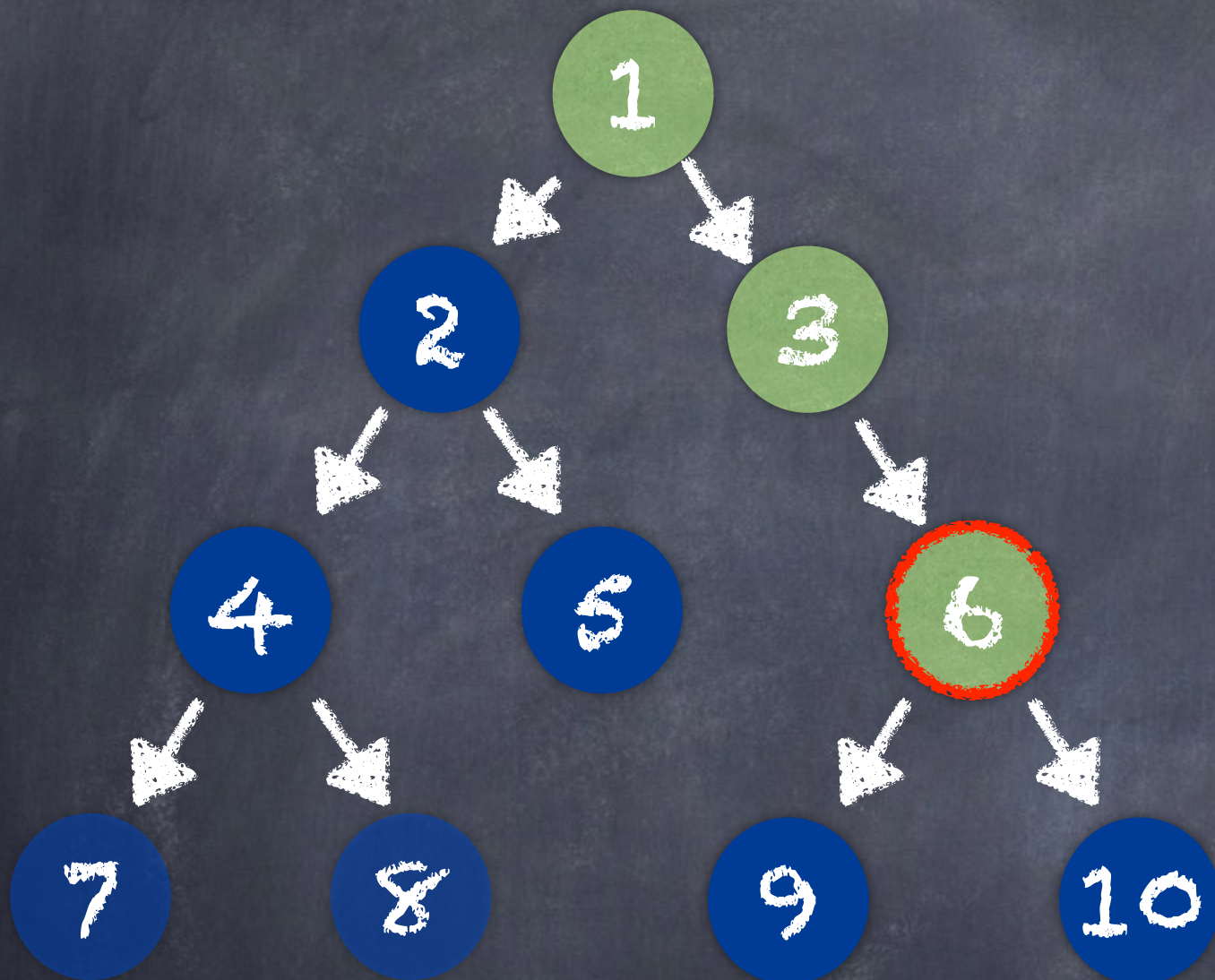
6

3

1

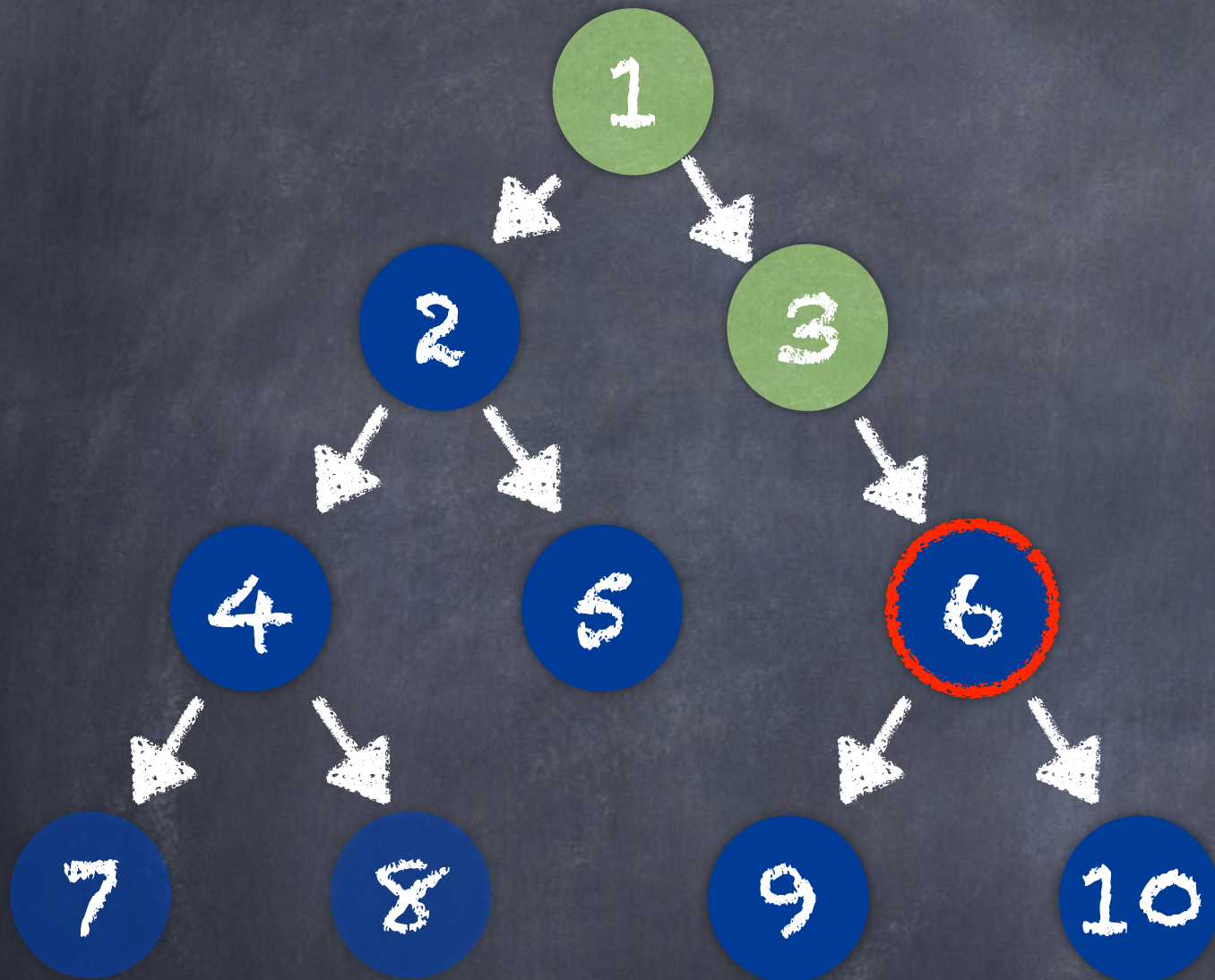
Pizzabestellung

Wiederholung
DFS



Pizzabestellung

Wiederholung
DFS

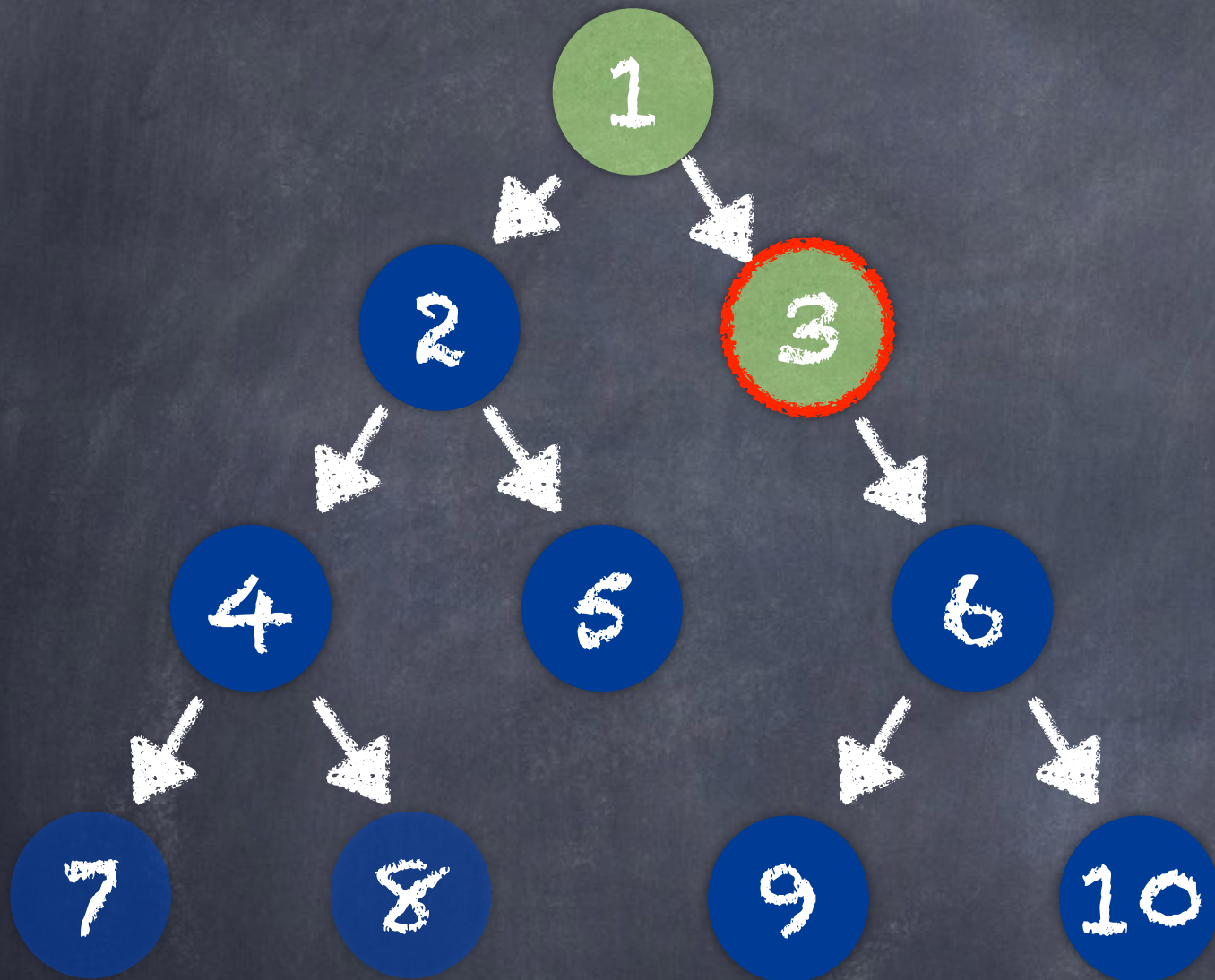


3

1

Pizzabestellung

Wiederholung
DFS

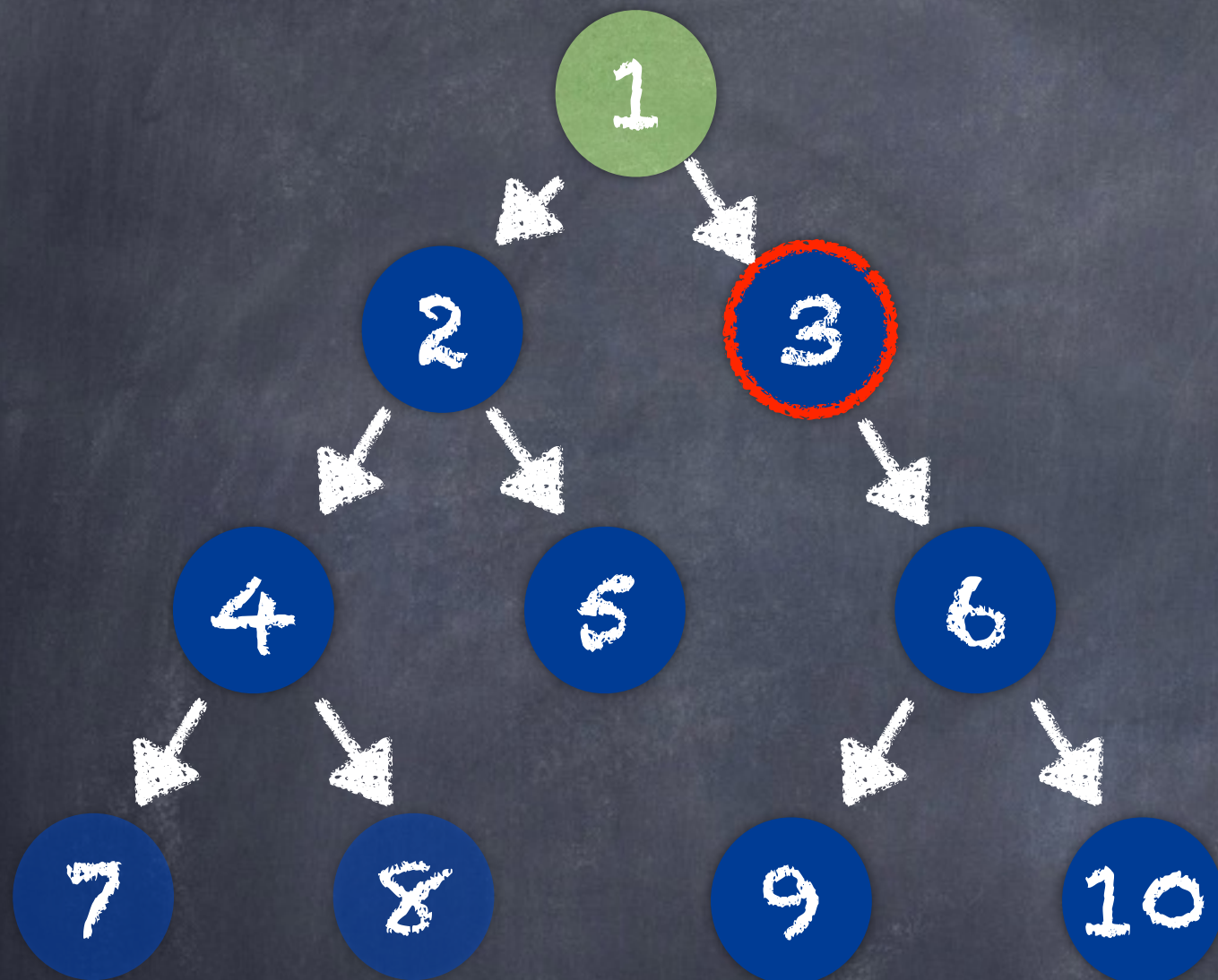


3

1

Pizzabestellung

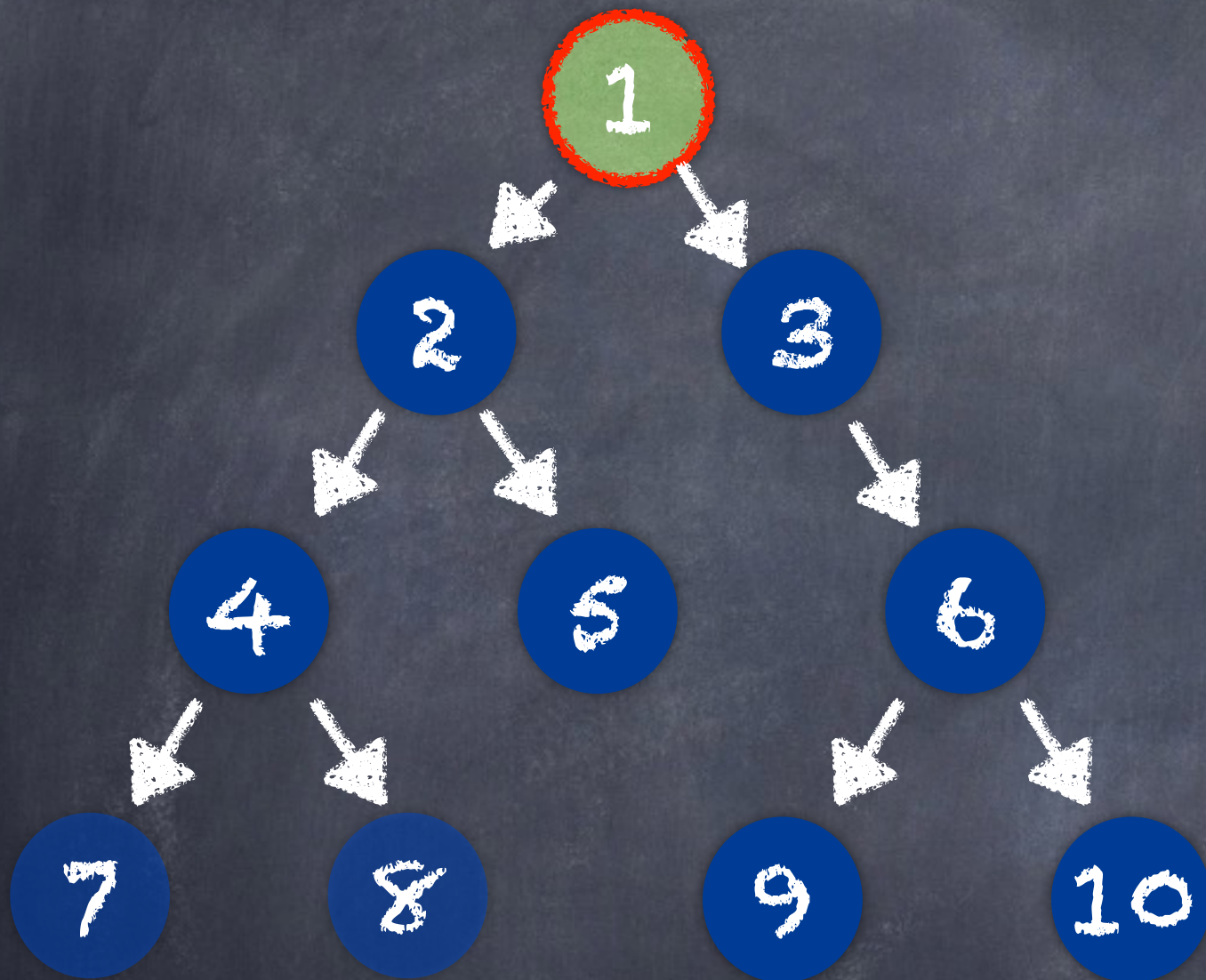
Wiederholung
DFS



1

Pizzabestellung

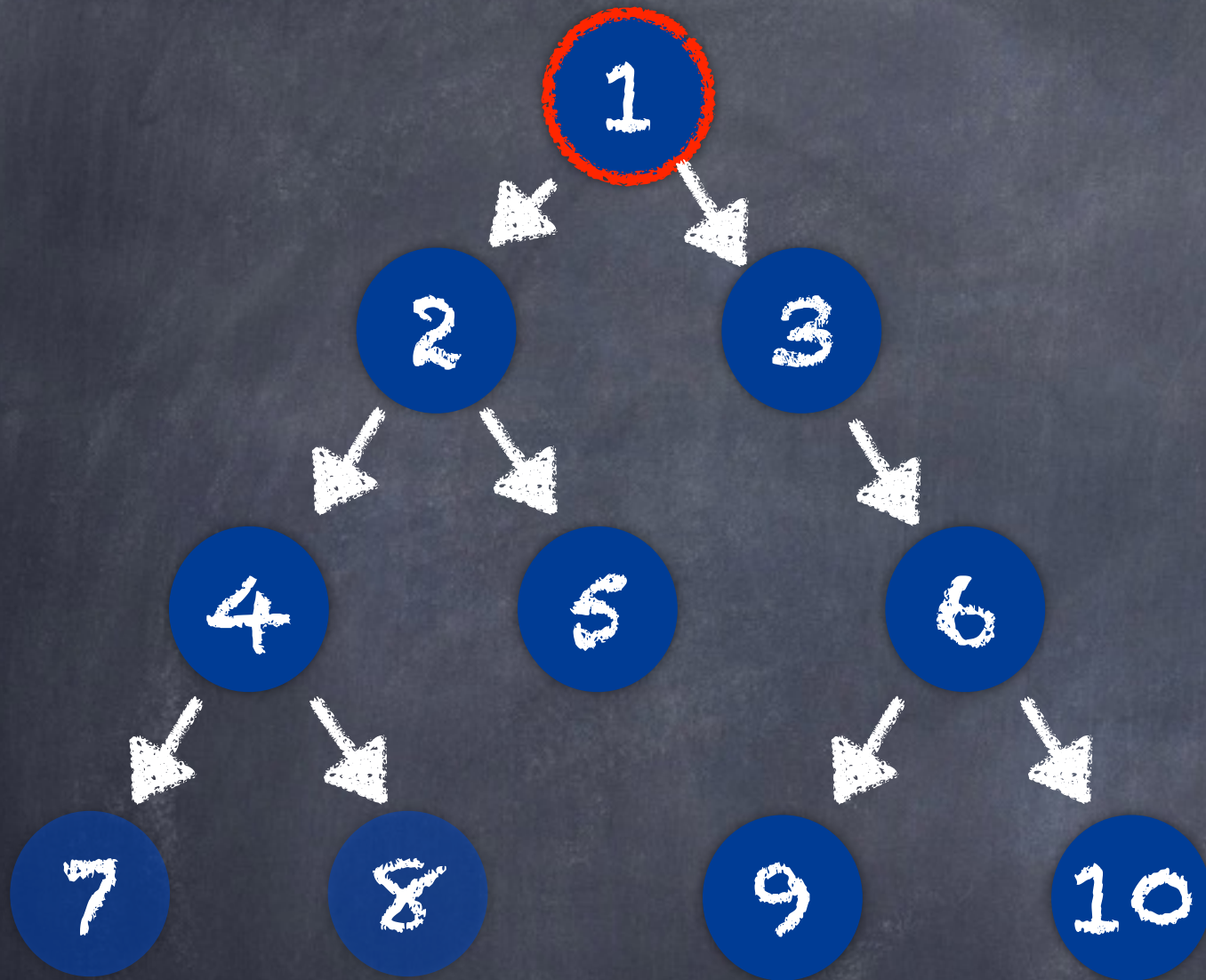
Wiederholung
DFS



1

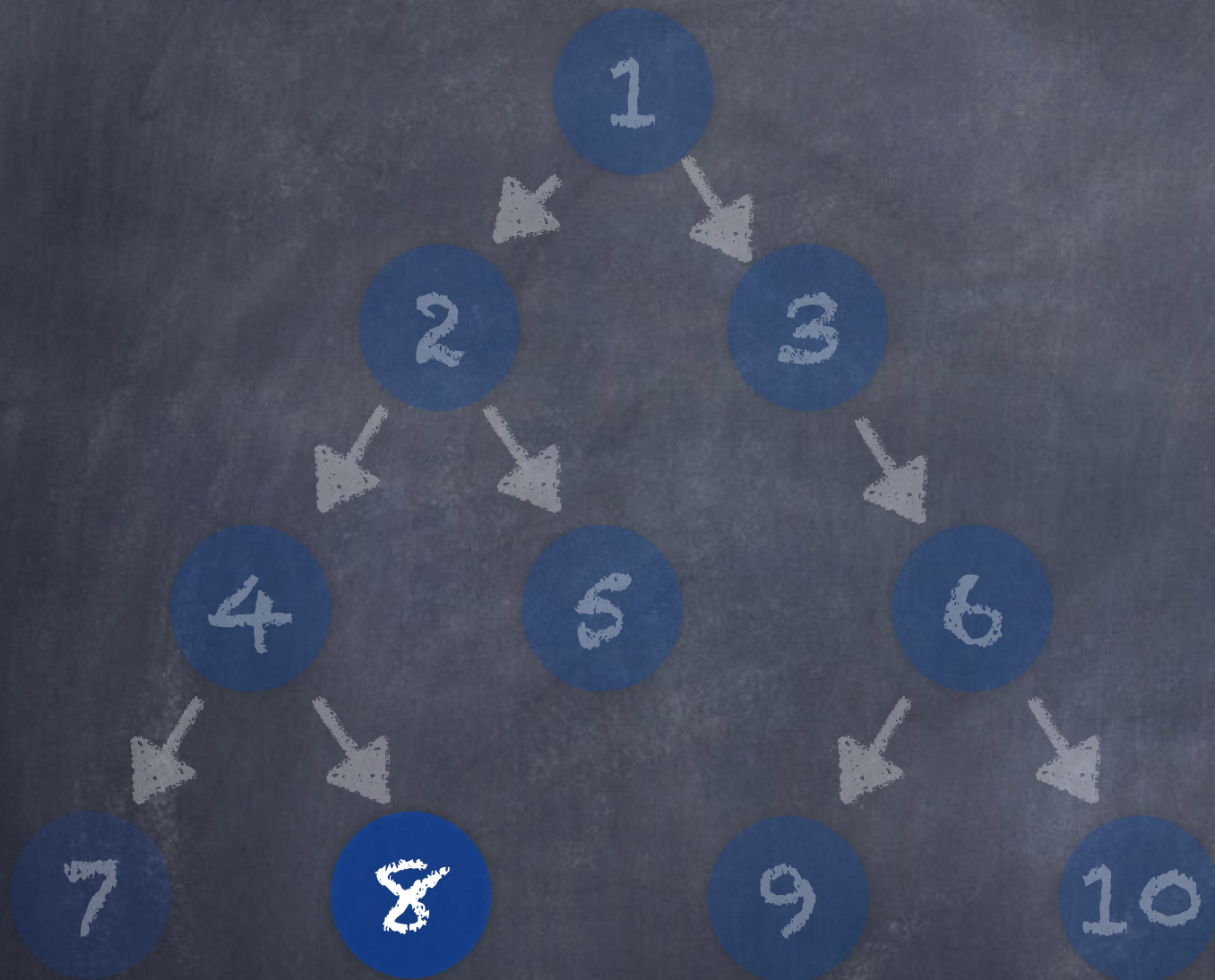
Pizzabestellung

Wiederholung
DFS



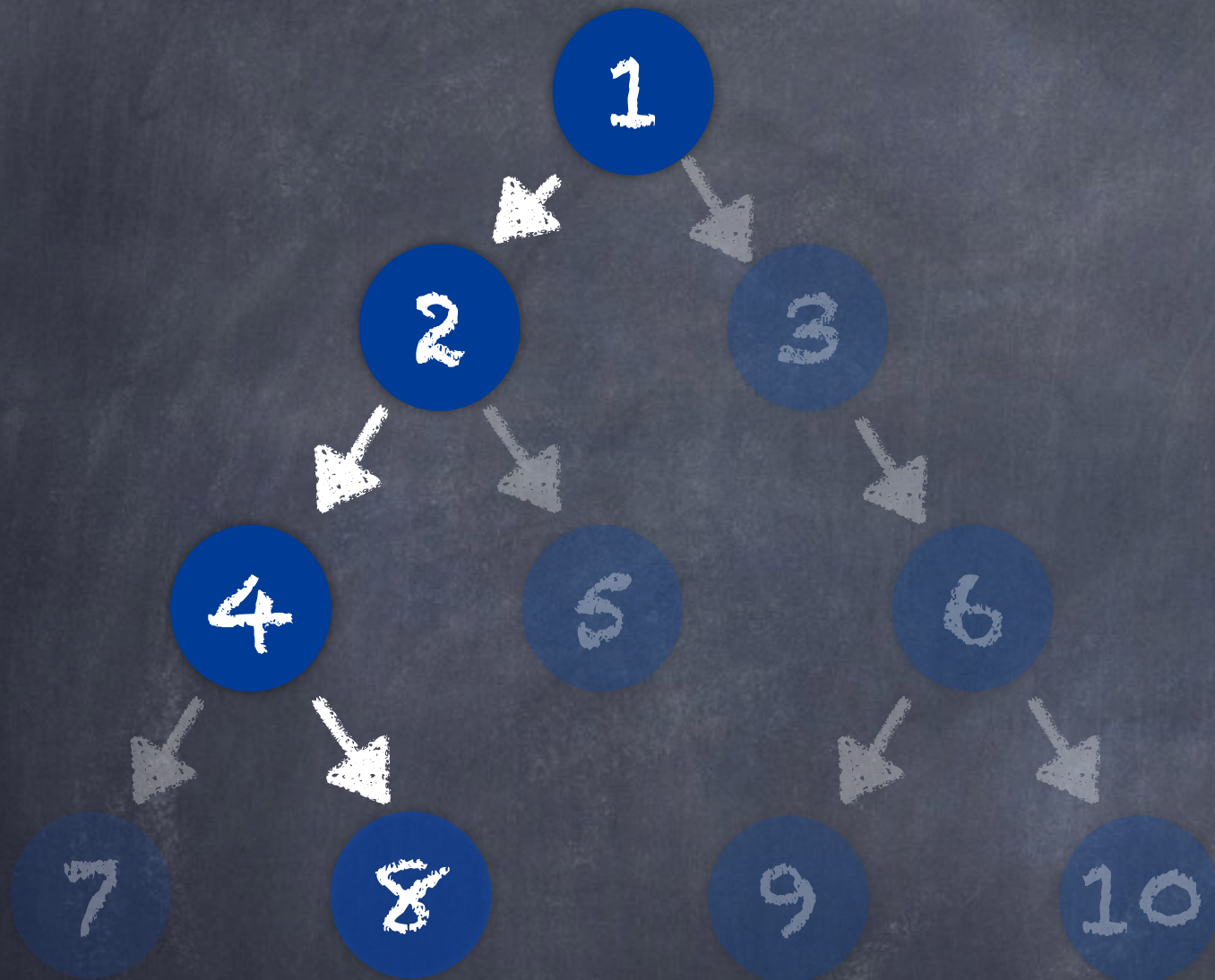
Pizzabestellung

Wiederholung
DFS

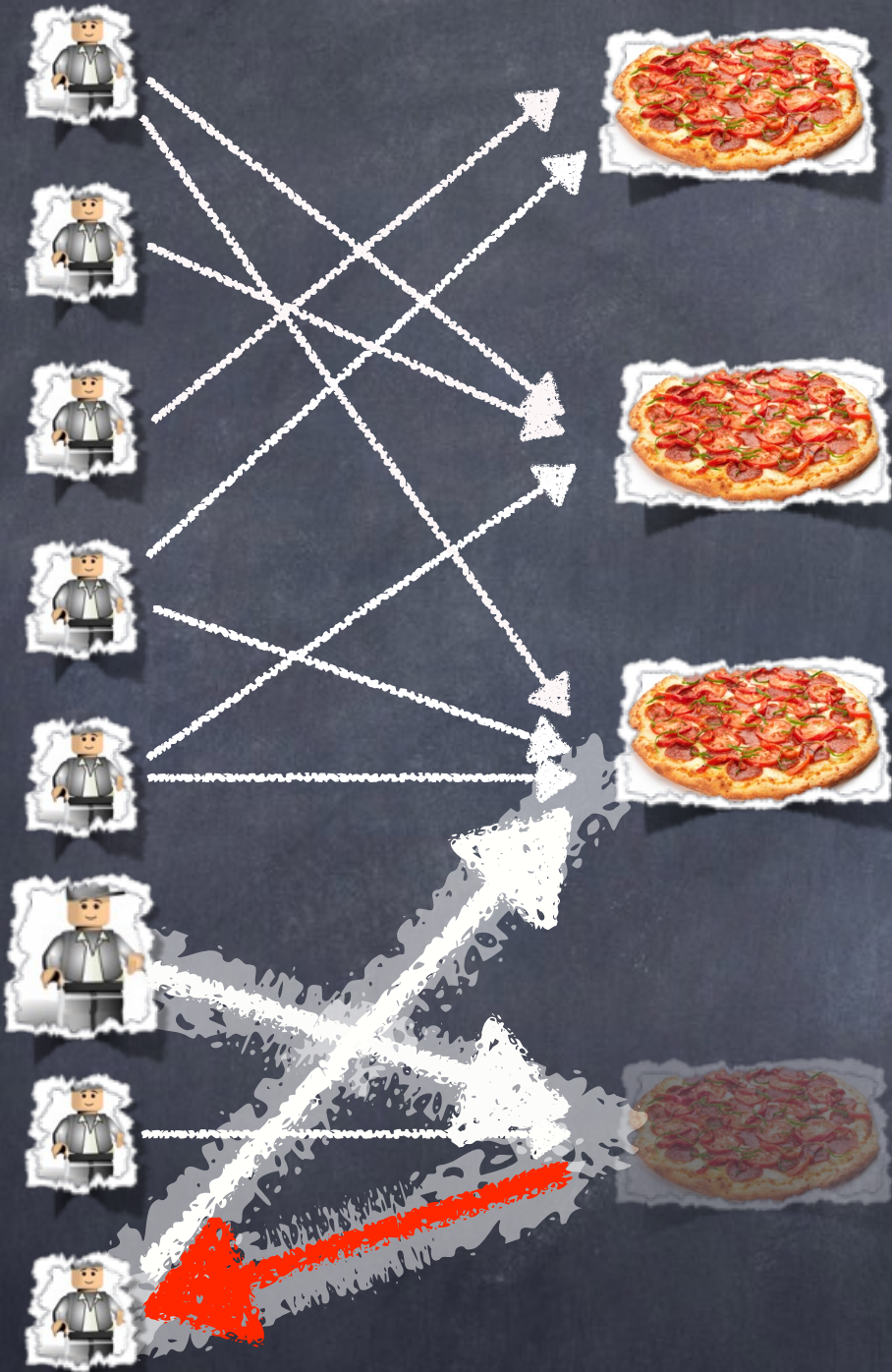


Pizzabestellung

Wiederholung
DFS



Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

$G' := (V, E')$

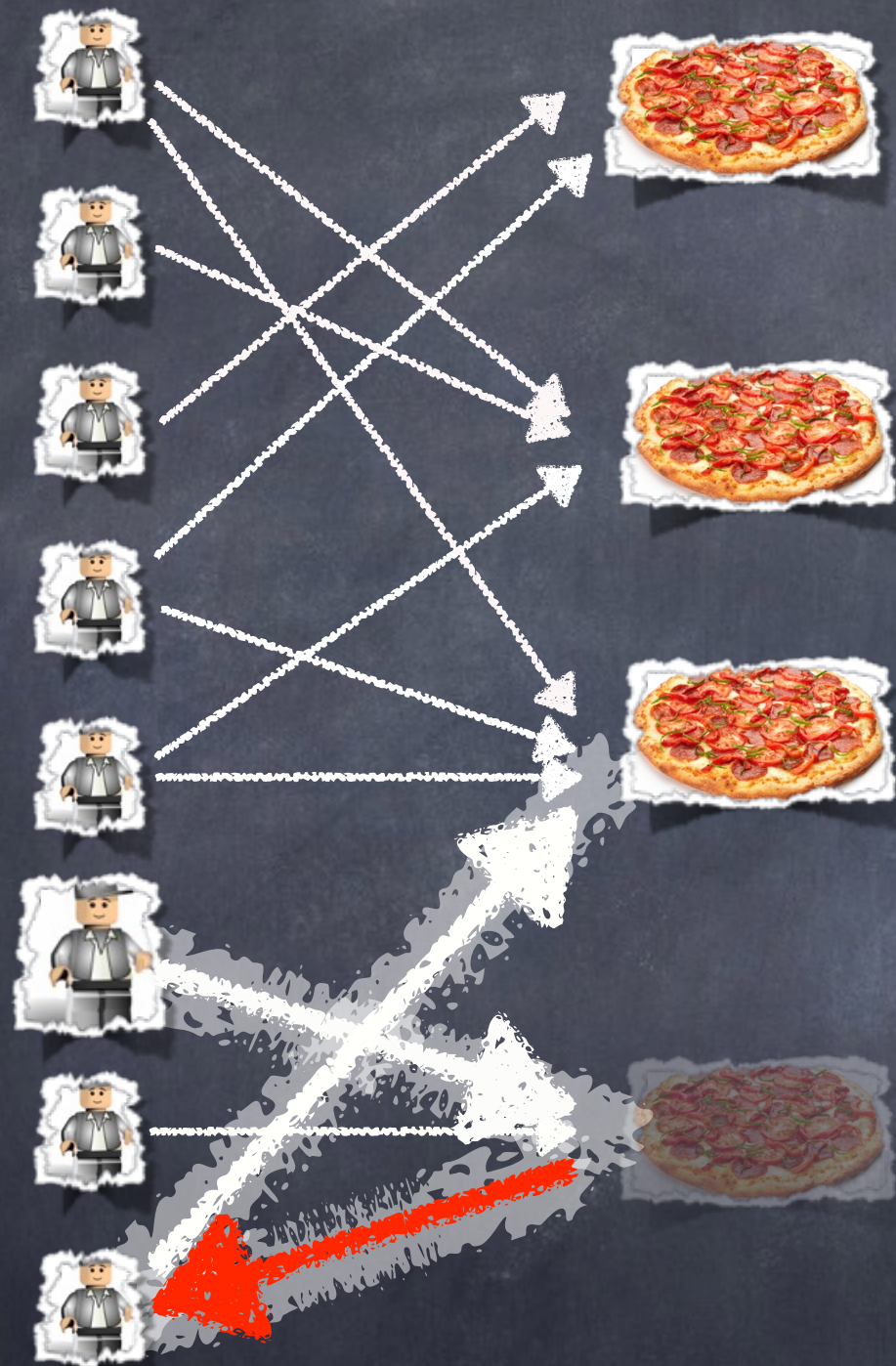
so lange möglich

$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

wie verbessern
wir E'

Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

$G' := (V, E')$

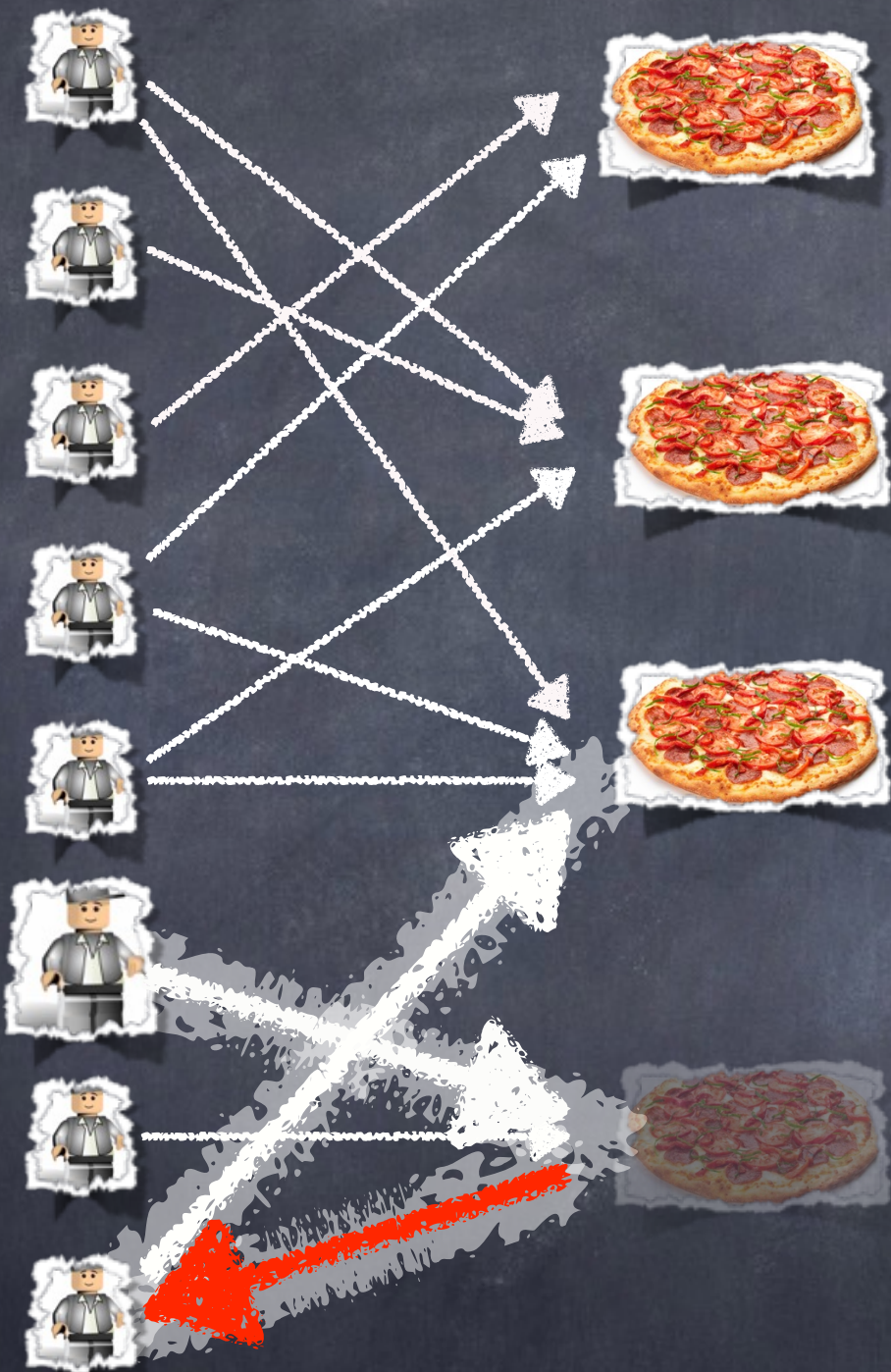
so lange möglich

$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

wie verbessern
wir E'

Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

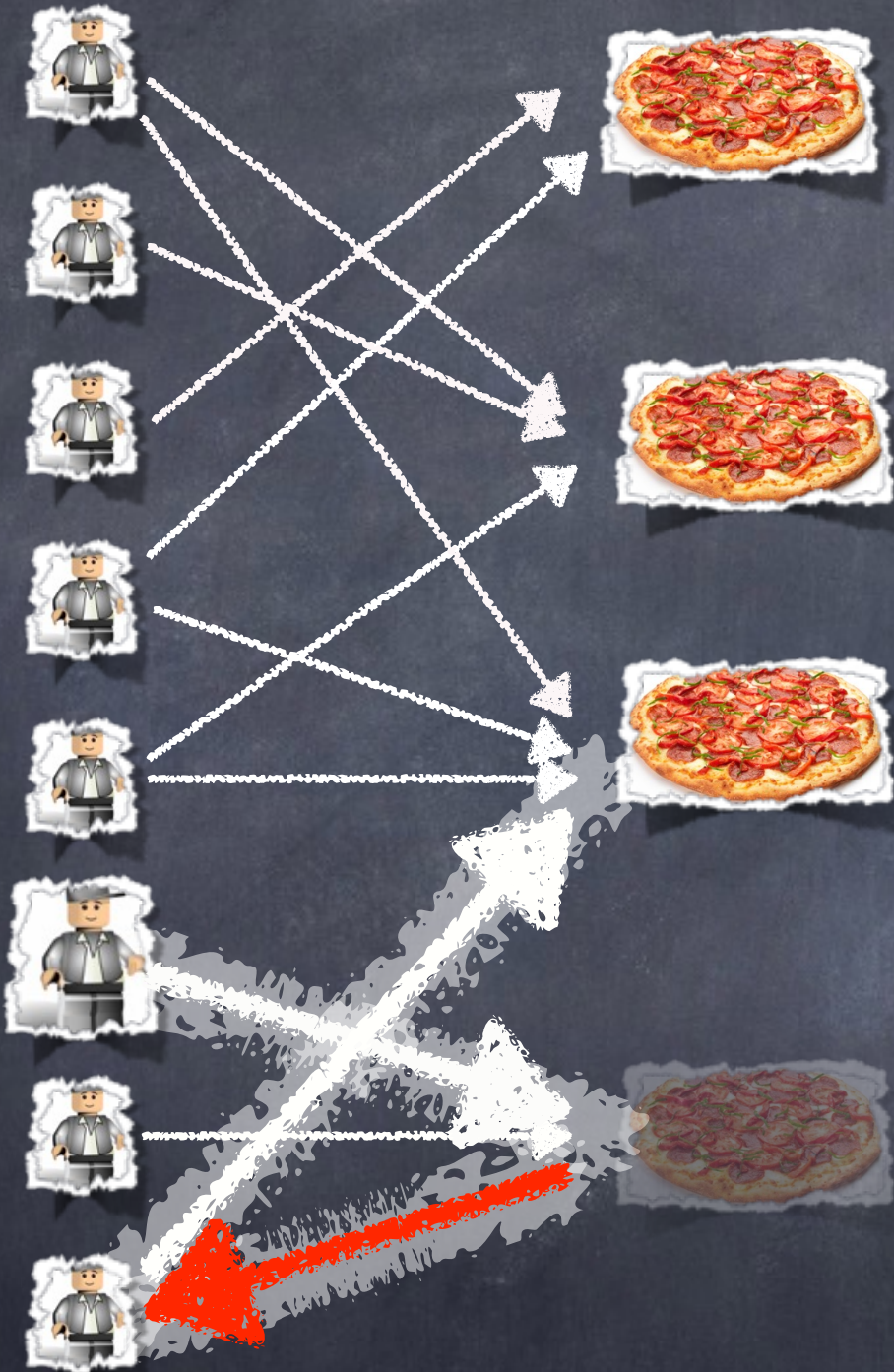
$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

wie verbessern
wir E'

Finde verbessernden
alternierenden Pfad

Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

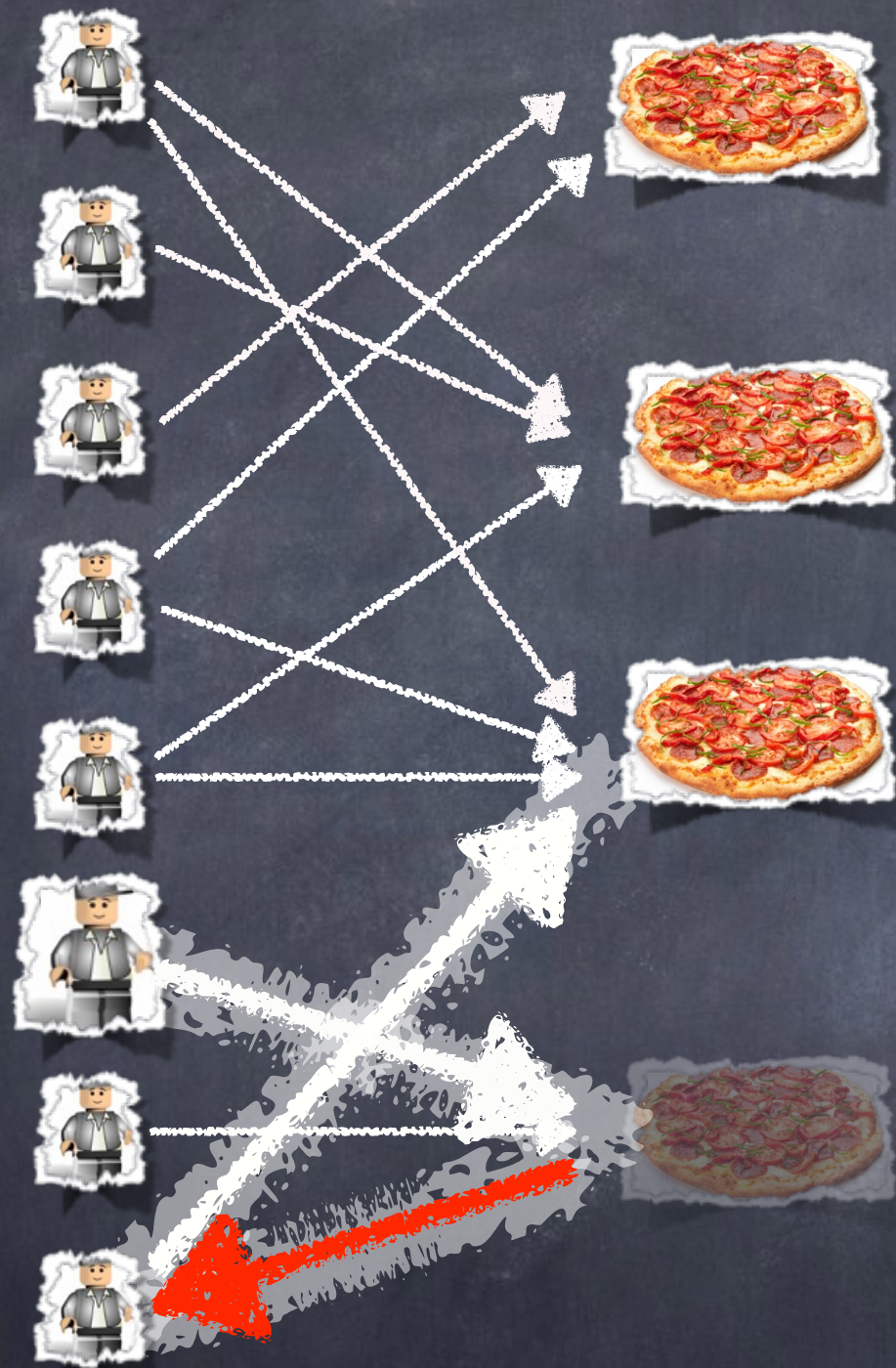
$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

$G' := (V, E')$

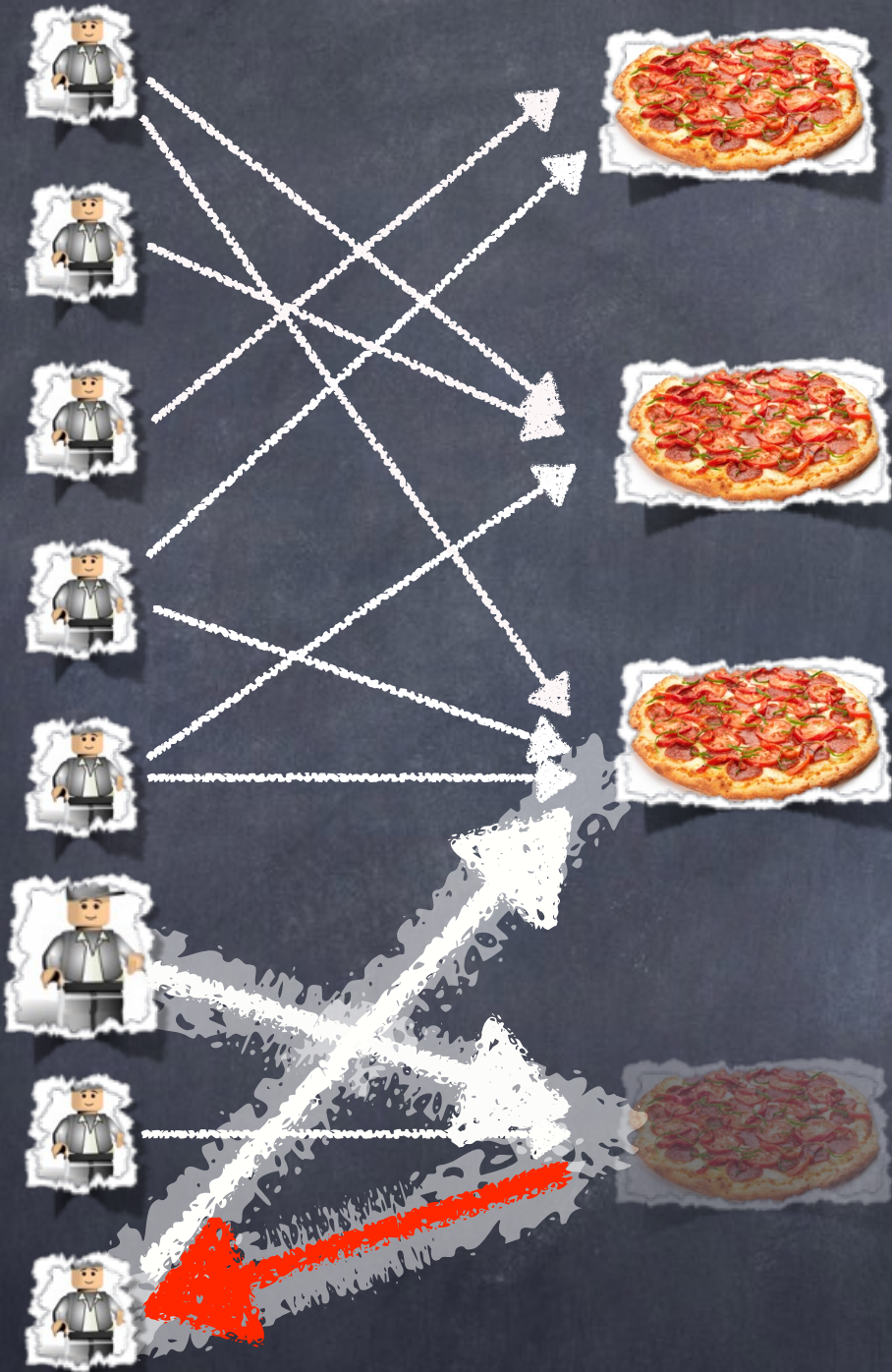
so lange möglich

$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

so lange BFS oder DFS
Pfad zu nicht überfülltem
Termin liefert

Pizzabestellung



Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

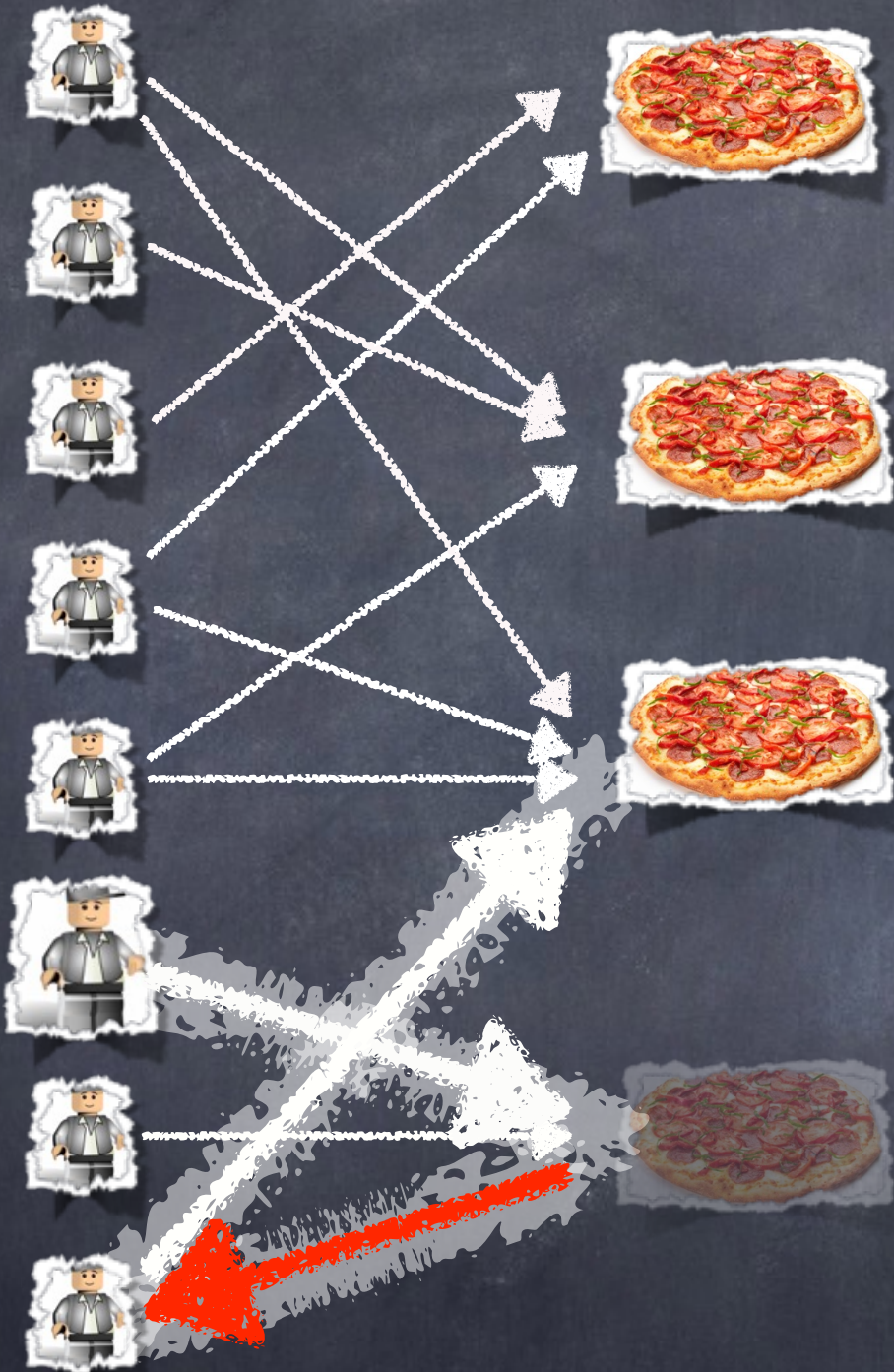
$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

so lange BFS oder DFS
Pfad zu nicht überfülltem
Termin liefert

→ solange Pizzaverteilung
um 1 besserbar

Pizzabestellung



$Pizza(G)$

$E' := \emptyset$

$G' := (V, E')$

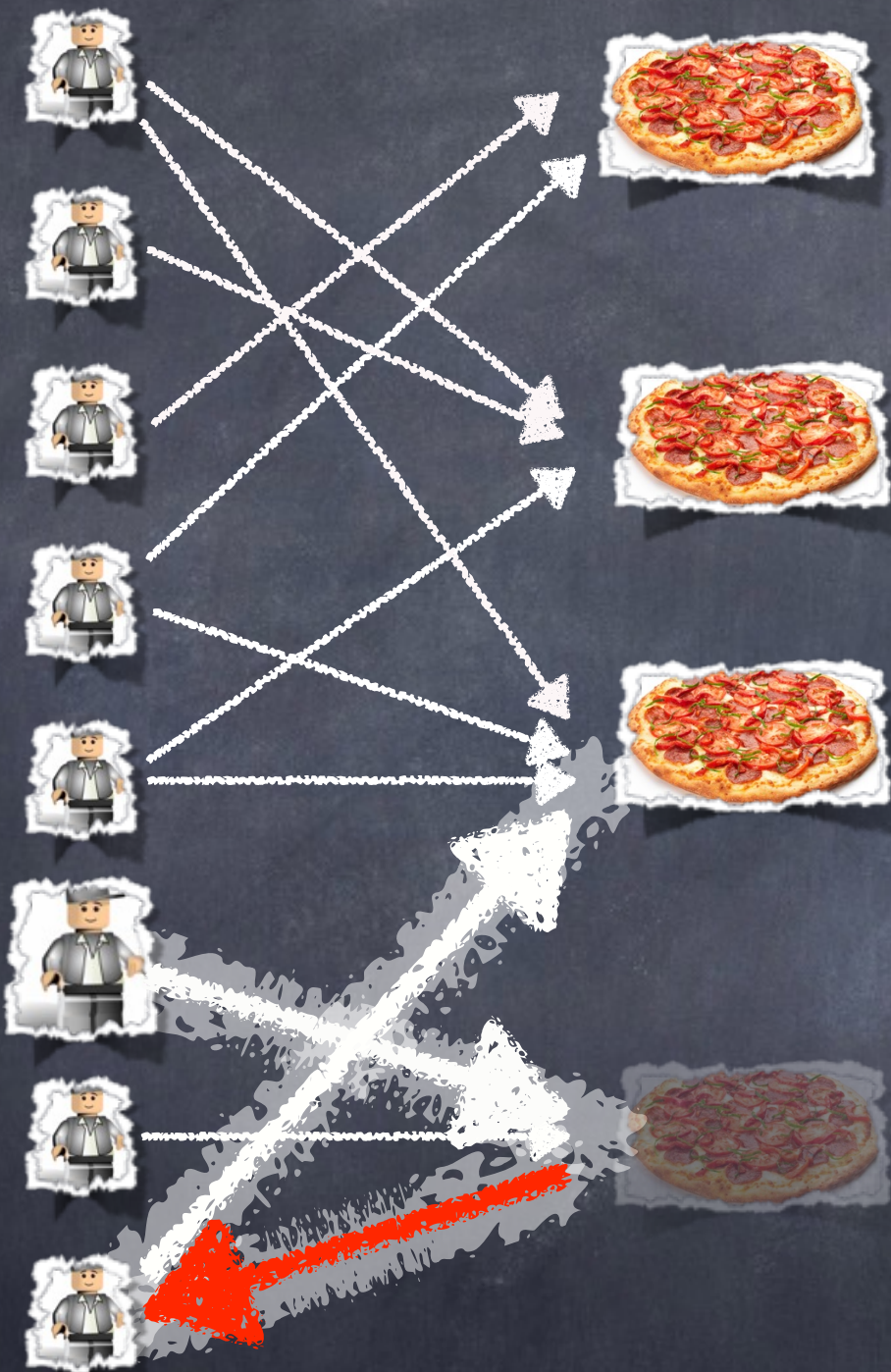
so lange möglich

$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

wie verbessern
wir E'

Pizzabestellung



Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

was heißt
„so lange möglich“?

wie verbessern
wir E'

Pizzabestellung

Analyse

Pizza(G)

$$E' := \emptyset$$

$$G' := (V, E')$$

so lange möglich

$$E' := \text{verbessere } E'$$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1
 \Rightarrow max. n Iterationen
warum? $|E'| \leq n$

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen
warum?

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen

warum?

Es gibt nur $30 \cdot m$ Stücke

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen

warum? $\forall v \in V_1 : \delta(v) \leq 30$

Es gibt nur $30 \cdot m$ Stücke

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen

warum? $\forall v \in V_1 : \delta(v) \leq 30$

Es gibt nur $30 \cdot m$ Stücke

\Rightarrow nicht mehr als $\max\{n, 30 \cdot m\}$ Iterationen

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen

warum? $\forall v \in V_1 : \delta(v) \leq 30$

Es gibt nur $30 \cdot m$ Stücke

\Rightarrow nicht mehr als $\max\{n, 30 \cdot m\}$ Iterationen

- Sei $T(G)$ benötigte Zeit für BFS/DFS auf G
also für eine Iteration

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen

warum? $\forall v \in V_1 : \delta(v) \leq 30$

Es gibt nur $30 \cdot m$ Stücke

\Rightarrow nicht mehr als $\max\{n, 30 \cdot m\}$ Iterationen

- Sei $T(G)$ benötigte Zeit für BFS/DFS auf G
also für eine Iteration

\Rightarrow Gesamt benötigte Zeit: $T(G) \cdot \min\{n, 30 \cdot m\}$

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen

warum? $\forall v \in V_1 : \delta(v) \leq 30$

Es gibt nur $30 \cdot m$ Stücke

\Rightarrow nicht mehr als $\max\{n, 30 \cdot m\}$ Iterationen

- Sei $T(G)$ benötigte Zeit für BFS/DFS auf G
also für eine Iteration

\Rightarrow Gesamt benötigte Zeit: $T(G) \cdot \min\{n, 30 \cdot m\}$

\Rightarrow Verfahren terminiert

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen

warum? $\forall v \in V_1 : \delta(v) \leq 30$

Es gibt nur $30 \cdot m$ Stücke

\Rightarrow nicht mehr als $\max\{n, 30 \cdot m\}$ Iterationen

- Sei $T(G)$ benötigte Zeit für BFS/DFS auf G
also für eine Iteration

\Rightarrow Gesamt benötigte Zeit: $T(G) \cdot \min\{n, 30 \cdot m\}$

\Rightarrow Verfahren terminiert

- Verfahren endlich beschreibbar

\Rightarrow Algorithmus

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

Pizzabestellung

Analyse

- pro Schleifeniteration
Verbesserung um 1

\Rightarrow max. n Iterationen

warum? $|E'| \leq n$

Nicht mehr als n Studenten einteilbar

\Rightarrow max. $30 \cdot m$ Iterationen

warum? $\forall v \in V_1 : \delta(v) \leq 30$

Es gibt nur $30 \cdot m$ Stücke

\Rightarrow nicht mehr als $\max\{n, 30 \cdot m\}$ Iterationen

- Sei $T(G)$ benötigte Zeit für BFS/DFS auf G
also für eine Iteration

\Rightarrow Gesamt benötigte Zeit: $T(G)$

\Rightarrow Verfahren terminiert

- Verfahren endlich beschreibbar

\Rightarrow Algorithmus

Pizza(G)

$E' := \emptyset$

$G' := (V, E')$

so lange möglich

$E' := \text{verbessere } E'$

