

Oppgave

Et kart konstrueres ved å trekke rette streker mellom ytterkantene. Hvordan kan vi fargelegge kartet med to farger?

Vis at algoritmen er korrekt.

Naboregioner kan ikke dele farge.

Vi har ingen identiske streker.

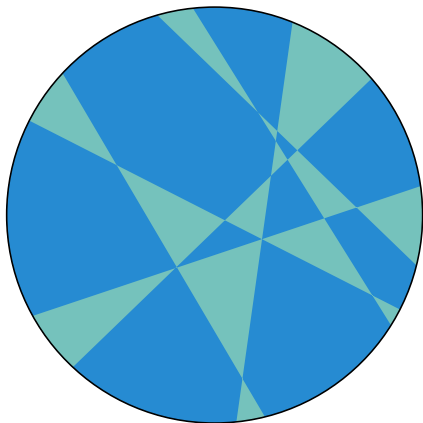
Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00



Oppgave

Et kart konstrueres ved å trekke rette streker mellom ytterkantene. Hvordan kan vi fargelegge kartet med to farger?

Vis at algoritmen er korrekt.

Naboregioner kan ikke dele farge.

Vi har ingen identiske streker.

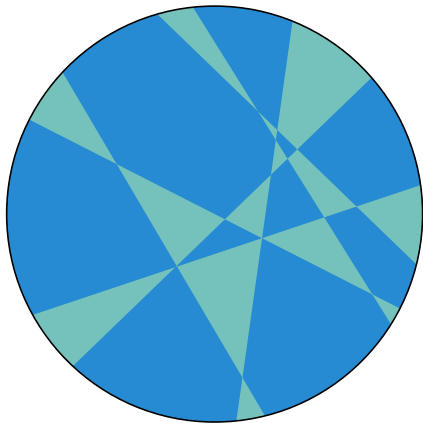
Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00



Oppgave

Et kart konstrueres ved å trekke rette streker mellom ytterkantene. Hvordan kan vi fargelegge kartet med to farger?

Vis at algoritmen er korrekt.

Naboregioner kan ikke dele farge.

Vi har ingen identiske streker.

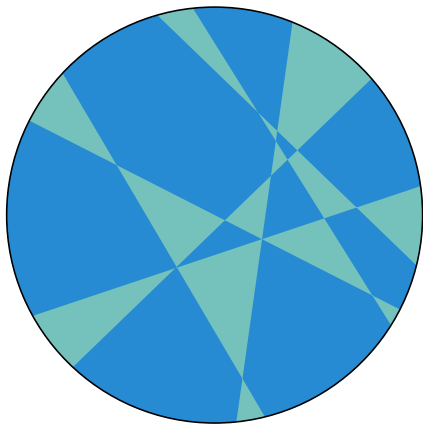
Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00



Oppgave

Et kart konstrueres ved å trekke rette streker mellom ytterkantene. Hvordan kan vi fargelegge kartet med to farger?

Vis at algoritmen er korrekt.

Naboregioner kan ikke dele farge.

Vi har ingen identiske streker.

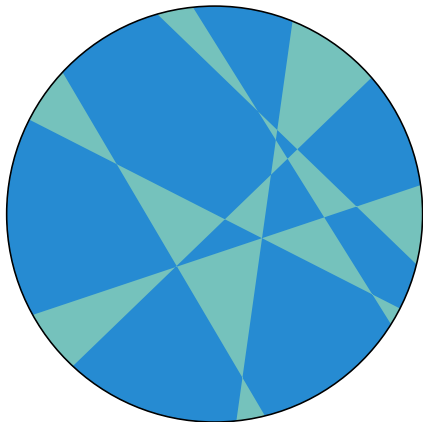
Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00



Oppgave

Et kart konstrueres ved å trekke rette streker mellom ytterkantene. Hvordan kan vi fargelegge kartet med to farger?

Vis at algoritmen er korrekt.

Naboregioner kan ikke dele farge.

Vi har ingen identiske streker.

Tenk selv 0:30

Jobb sammen 1:30

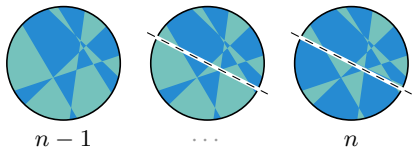
Observasjoner

Løsningsforslag

Refleksjon 1:00

Løsningskisse

Induksjon over streker. Løs for $n - 1$. Legg til strek n . Naboer på hver sin side av streken har samme farge; bytt alle farger på den ene siden.



Vi kan fargelegge én og én region. Mer effektivt, og opplagt rett når to farger er nok, men vi må vise at det *er* nok.

Oppgave

Et kart konstrueres ved å trekke rette streker mellom ytterkantene. Hvordan kan vi fargelegge kartet med to farger?

Vis at algoritmen er korrekt.

Naboregioner kan ikke dele farge.

Vi har ingen identiske streker.

Tenk selv 0:30

Jobb sammen 1:30

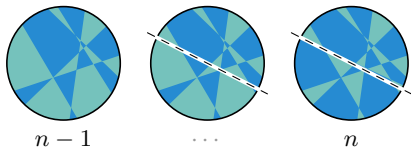
Observasjoner

Løsningsforslag

Refleksjon 1:00

Løsningskisse

Induksjon over streker. Løs for $n - 1$. Legg til strek n . Naboer på hver sin side av streken har samme farge; bytt alle farger på den ene siden.



Vi kan fargelegge én og én region. Mer effektivt, og opplagt rett når to farger er nok, men vi må vise at det *er* nok.

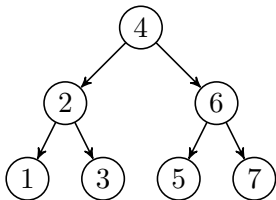
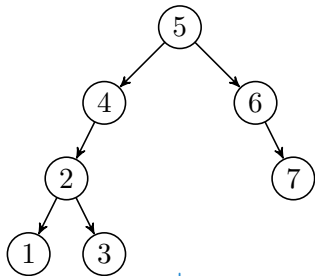
Hva tenkte og gjorde du? Hvorfor? Hva fungerte? Glemte du noe? Hva skjønner du nå? Hvilke nye sammenhenger ser du? Hva skjønner du fortsatt ikke? Hva vil du huske på eller gjøre annerledes senere?

Oppgave

Hvordan kan du balansere et binært søketre i lineær tid?

Hint: Kan du bygge et balansert binært søketre fra en sortert tabell?

Tenk selv	0:30
Jobb sammen	1:30
Observasjoner	
Løsningsforslag	
Refleksjon	1:00



Oppgave

Hvordan kan du balansere et binært søketre i lineær tid?

Hint: Kan du bygge et balansert binært søketre fra en sortert tabell?

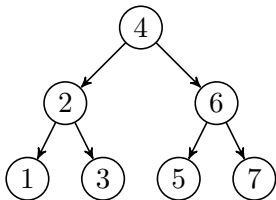
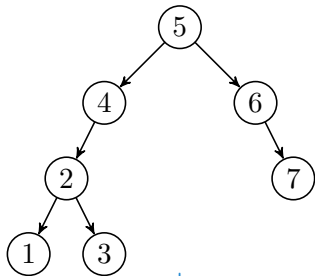
Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00



Oppgave

Hvordan kan du balansere et binært søketre i lineær tid?

Hint: Kan du bygge et balansert binært søketre fra en sortert tabell?

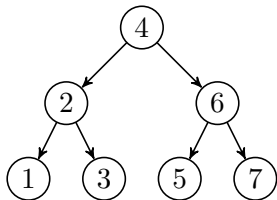
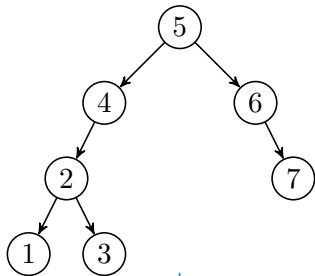
Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00



Oppgave

Hvordan kan du balansere et binært søketre i lineær tid?

Hint: Kan du bygge et balansert binært søketre fra en sortert tabell?

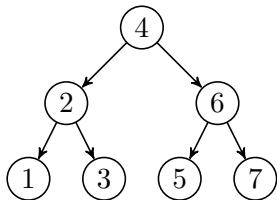
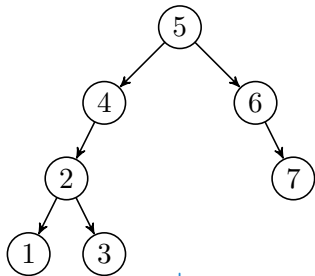
Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00



Oppgave

Hvordan kan du balansere et binært søketre i lineær tid?

Hint: Kan du bygge et balansert binært søketre fra en sortert tabell?

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

Løsningskisse

Traverser treet (*inorder*) for å få en sortert tabell A.

Midterste element blir rot; bygg venstre og høyre deltre rekursivt.

$$T(n) = 2T(n/2) + 1 \implies T(n) = \Theta(n)$$

Det finnes også *in-place*-løsninger.

Oppgave

Hvordan kan du balansere et binært søketre i lineær tid?

Hint: Kan du bygge et balansert binært søketre fra en sortert tabell?

Tenk selv 0:30

Jobb sammen 1:30

Observasjoner

Løsningsforslag

Refleksjon 1:00

Løsningskisse

Traverser treet (*inorder*) for å få en sortert tabell A.

Midterste element blir rot; bygg venstre og høyre deltre rekursivt.

$$T(n) = 2T(n/2) + 1 \implies T(n) = \Theta(n)$$

Det finnes også *in-place*-løsninger.

Hva tenkte og gjorde du? Hvorfor? Hva fungerte? Glemte du noe? Hva skjønner du nå? Hvilke nye sammenhenger ser du? Hva skjønner du fortsatt ikke? Hva vil du huske på eller gjøre annerledes senere?