

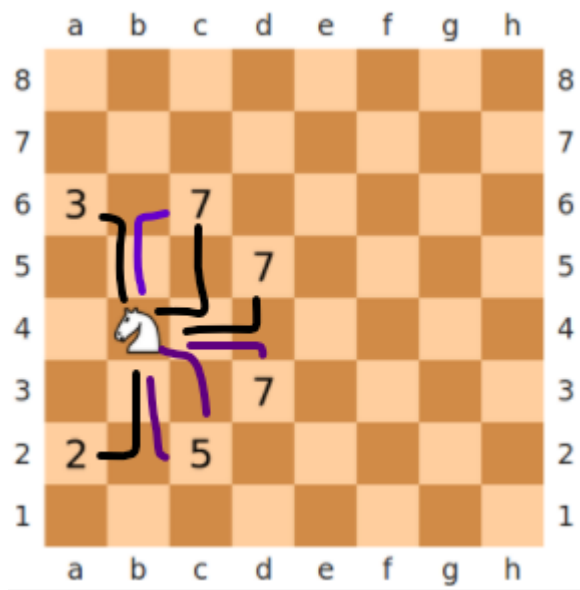
## Springerproblemet

Standardløsning :

1. «knight»/ hest har 8 mulige veier:

Valg mellom: går to opp også en høyre/venstre eller to opp og en ned høyre/venstre.

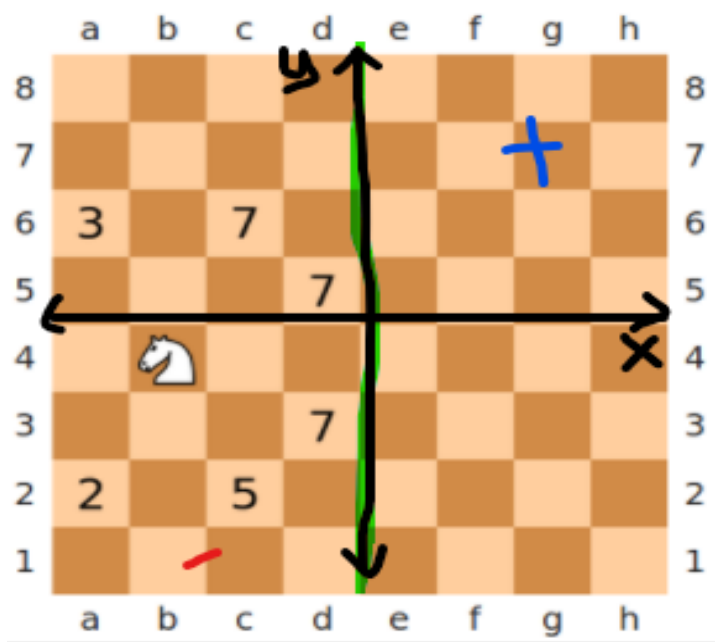
2. Rekursiv - Backtracking, ikke gå tilbake til steget man har allerede hvert på - stepCount
3. Ikke gå ut av brettspillet ( ved hjelp av sjekk)



- Som man kan se ved å telle de strekene, har man 8 mulige veier.
- Tenk det som akse. Hvis man går mot høyre er det positivt [1, 2 osv]
- Negativt y-akse er mot venstre.

## Springerproblemet

- Vertikal (grønt) : det som er ovenfor er positivt og det som er under er negativt



- -1, å ta en steg ned
- 1, å ta en steg opp
- 2, å ta to steg opp
- -2 å ta to steg ned.

### I Algoritme. Java

- Her skal man lagre referanse som `int [][]` som `Board ( brett) / Matrix`
- Definere stegene for hesten gjennom x og y akse, 1 dimensjonal – 8 mulige valg.
- I konstruktør instansierer brettspillet
- Lager ny klasse for konstanter, som privat og en public .
- Rekursiv – `stepCount`