

Aflevering 3
Algoritmer og datastrukturer
Forår/Sommer 2011

Naja Mottelson (vsj465)
Søren Pilgård (vpb984)

9. juni 2011

Indhold

1	3
---	---

1

Til at finde de konveksehyldster lag kan vi udnytte jarvis march-algoritmen. Vi ved at alle punkter skal traverseres og bruges som knuder. Hvis et punkt ikke bruges i et konvekst hylster vil det blive brugt i et af de følgende. I vores implementation godtager vi at det inderste lag kan bestå af 1 eller 2 punkter.

I vores algoritme bruger vi jarvis-march til at finde hvert konveks-hylster lag. Dette gentages så på den resterende mængde punkter, indtil alle punkterne er i et lag.

Vi har så at hvert punkt i mængden Q vil ligge som knude i et lag. Hvert punkt bruges kun 1 gang.

Vi ved at jarvis march algoritmen kører i $\theta(n*h)$ hvor n er antallet af punkter og h er antallet af knuder. Vi ved at efter hver iteration bliver n , h punkter mindre. Samtidig ved vi at alle punkter er knude i et lag, så $h = n$ og vi kan se at køretiden må være $\theta(n^2)$

Grundlæggende ved vi at hver jarvis-march iteration kigger på en knude og vinklerne til alle de resterende punkter. Da alle punkterne er knude i et lag, ser vi at jarvis-march skal bruges $h=n$ gange ergo må køretiden være $\theta(n * n)$

```
convex-layers(Q)
  let S be an empty list

  while Q is not empty
    if lenght(Q) >= 3:
      k = jarvis-march(Q)
    else:
      k = Q //there is only 1 or 2 points left
    Q = Q-k
    S.append(k)

  return reverse(S)
```