

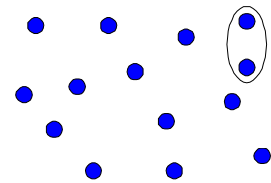
Alg.dat øving 9

Nærmeste punkt-par
&
0-1-ryggsekk problem

1

Nærmeste punktpar

- Problem:
 - Finne de 2 nærmeste punktene i en mengde
- Lengdemål
 - Manhattan-avstand (L_1 -avstand)



2

Anvendelser av nærmeste punktpar-algoritmen

- Trafikk-kontroll
 - Flytrafikk
- Klassifisering (clustering)
- Byggestein for andre algoritmer

3

Løsningside

- Generelt: Rekursjon – splitt og hersk
- Idé:
 - Deler punktmengden P i 2 like deler:
 - $P_{venstre}$ og $P_{høyre}$ og løser rekursivt
 - Må ha et av tilfellene:
 1. Nærmeste punktpar ligger i $P_{venstre}$
 2. Nærmeste punktpar ligger i $P_{høyre}$
 3. Et punkt ligger i $P_{venstre}$ og et i $P_{høyre}$

4

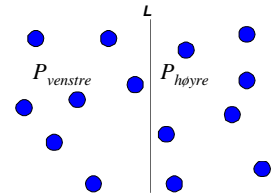
Initiering og rekursjonskriterier

- **Initiering:**
 - Lag sorterte tabeller X og Y av punkter i P
 - X inneholder hele P sortert etter x-koordinat
 - Y inneholder hele P sortert etter y-koordinat
 - Kjøretid: $O(n \lg(n))$
- **Terminering og rekursivt kall**
 - $|P| \leq 3$ – gjør alle til alle sjekk
 - Hvis ikke, gjør rekursivt kall

5

Splitt

1. Lag $P_{venstre}$ og $P_{høyre}$
 - Like mange punkter i hver mengde
2. Lag $X_{venstre}$ og $X_{høyre}$ basert på $P_{venstre}$ og $P_{høyre}$
3. Lag $Y_{venstre}$ og $Y_{høyre}$ basert på $P_{venstre}$ og $P_{høyre}$



6

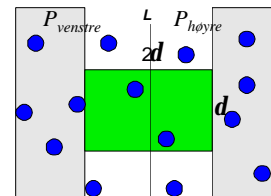
Hersk

- **Rekursivt kall:**
 1. $d_{venstre} = f(P_{venstre}, X_{venstre}, Y_{venstre})$
 2. $d_{høyre} = f(P_{høyre}, X_{høyre}, Y_{høyre})$
- **Finn det nærmeste par avstand**
 - $d = \min(d_{venstre}, d_{høyre})$
 - **MÅ** sjekke om det finnes et nærmere par med et punkt i hver av mengdene $P_{høyre}$ og $P_{venstre}$

7

Kombiner

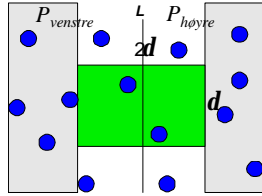
1. Lag $Y' \hat{=} Y$
 - Ta bort områder til venstre og høyre
2. Lag glidende rektangel $2\delta * \delta$ stort
3. Gå igjennom punktene i Y' og sjekk mot rektangelområdet



8

Geometrisk forståelse I

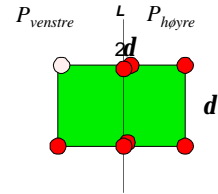
1. Skal finne 2 punkter, ett i $P_{venstre}$ og ett i $P_{høyre}$ som har mindre enn d avstand
 - Kan fjerne punkter til høyre og venstre for rektangel (for langt fra til å være kandidat)
 - For et kandidatpunkt (på topprand av rektangel), trenger man kun å se på kandidater i rektangelet (høyde d)



9

Geometrisk forståelse II (Euklidsk avstand)

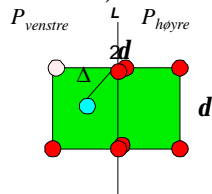
2. Max antall medkandidater per fiksert rektangel er 7 (langs randen)
3. Grunn:
Hvis punktet lå inne i rektangelet ville man fått et annet resultat for d



10

Geometrisk forståelse III (Euklidsk avstand)

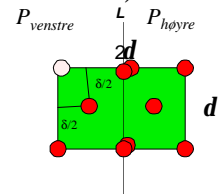
4. Her ville man fått en annen d (vist som Δ på figur)
 - mindre rektangel og begynne på nytt



11

Geometrisk forståelse IV (Manhattan avstand)

- Kan ha maks 9 aktuelle punkter grunnet at Manhattan avstand geometrisk er først å følge den ene aksen og deretter den andre i mots $\delta/2$



(gå rundt kvartal i NYC)

12

0-1 Knapsack - Problem

- Mål: Maksimer verdien av sekkens innhold
- Har:
 - n gjenstander ($1 \leq i \leq n$)
 - Hver gjenstand har:
 - Verdi v_i
 - Vekt w_i
- Begrensning:
 - Kan kun løfte en masse på W totalt.

13

0-1 Knapsack - Formelt

- Uttrykk:

$$V(i, j) = \max_S \left\{ \sum_{k \in S} v_k \mid k \leq i, \sum_{k \in S} w_k \leq j \right\}$$

- $v_k = 0 \Leftrightarrow$ gjenstand k ikke tatt med
- $w_k = 0 \Leftrightarrow$ gjenstand k ikke tatt med
- j er kapasiteten til delproblemet (W')

14

0-1 Knapsack – Initiering

Har:

- $V(i, 0) = 0$
 - 0 i maksverdi når kapasiteten er 0
- $V(0, j) = 0$
 - 0 i maksverdi når tyngste gjenstand har tyngde 0

15

0-1 Knapsack – Løsning&forklaring

Formelt:

$$V(i, j) = \begin{cases} V(i-1, j) & \text{hvis } j < w_i \\ \max\{V(i-1, j), V(i-1, j-w_i) + v_i\} & \text{ellers} \end{cases}$$

Forklaring:

1. Hvis w_i tyngre enn total kapasitet, ikke ta med w_i

16

0-1 Knapsack – Løsning&forklaring

Formelt:

$$V(i, j) = \begin{cases} V(i-1, j); & \text{hvis } j < w_i \\ \max\{V(i-1, j), V(i-1, j-w_i) + v_i\}; & \text{ellers} \end{cases}$$

Forklaring:

2. Velg max av løsning med å ikke ta med w_i og av å ta med w_i

– NB: Hvis man tar med w_i blir kapasiteten redusert, derfor j-