5. Randomized Algorithms Sources • Chapter 13 • November 13 part 1

Intro Normalt kan vi tænke på random og algoritmer på to forskellige måder. Vi kan enten tænke på det som at vi får tilfældigt input ude fra, altså vi antager at det er ligegyldigt og forsøger at lave en avarage case running time analyse på det. En anden måde at se det på er ved at undersøge randomized algoritmer hvor vi antager et worst case scenario og i stedet for algoritmen til at opfører sig tilfældigt for at undgå disse cases.

Randomized Median Finder Givet at vi gerne ville finde medianen i et sæt af usorterede tal, ville en simpel løsning være at sortere input først og så tage det midterste tal. Problemet med denne deterministiske løsning, er at det tager

$$\theta(n \cdot log(n))$$

. Hvad hvis vi gerne vil gøre det i lineær tid?

Lad os nummere elementerne i sættet hvor medianen eksistere

$$S = \{a_0, a_1, ..., a_j, ..., a_{n-1}, a_n\}, n = 2k + 1, k \in \mathbb{N}$$

.

Vi ønsker nu at kunne finde element nummer

$$r$$

givet et sæt var sorteret i et sæt

$$L$$

. Lad os denotere denne rutine som

$$M(L, r - 1)$$

. Vi vælger nu en tilfældig pivot

$$a_j \in S$$

og itterære nu over alle

$$a_i \in S \setminus \{a_j\}$$

og sortere dem i to sæt sådan at

$$S^+ = \{a_i | a_i \in S \setminus \{a_j\} \wedge a_i > a_j\}$$

$$S^- = \{a_i | a_i \in S \setminus \{a_j\} \wedge a_i < a_j\}$$

Vi kan nu finde ud af hvilket sæt vi skal kigge i for at finde medianen, vi har tre cases

$$|S^-| = r \rightarrow halt$$

$$|S^-| < r \rightarrow M(S^+, r - |S^-|)$$

$$|S^-| > r \rightarrow M(S^-, r)$$

Givet at vi kan lave de to sæt i linær tid, så har vi en køretid på

$$T(n) = c \cdot n + T(h), h < n$$

Antag nu worstcase, hvor hvert split vi laver med vores pivot

$$a_j$$

kun laver et split det gør subset én mindre sådan at

$$T(n) = c \cdot n + T(n-1)$$

, i så fald får vi en køretid på

$$T(n) \leq c \cdot n + c \cdot (n-1) + c \cdot (n-2) + ... = \frac{c \cdot n \cdot (n+1)}{2} = \theta(n^2)$$

Så vi bliver nødt til at lave gode splits. Antag at vi laver splits der reducerer hvert recursivt kald, i så fald får vi en køretid på

$$T(n) = c \cdot n + T((1-\epsilon) \cdot n) = \theta(n)$$

Så hvordan laver vi et sådan split? Vi definiere en god pivot

$$a_j$$

sådan

$$|L^-| \geq \frac{1}{4} \cdot n \wedge |L^+| \geq \frac{1}{4} \cdot n$$

Vi kan se at der er 50

$$E[X_j] \leq 2c \cdot n \cdot (\frac{3}{4})^j$$

Hvis vi summere over alle rekusioner sådan at j går mod uendelig, så har vi at

$$E(X) = \sum_{j=0}^{\infty} 2c \cdot n \cdot (\frac{3}{4})^j \leq 8cn$$

Vi konkluderer derfor linær kørertid ved at bruge randomization.