

Komplexiteten för binär sökning

Binär sökning

Problem

Det finns n element i en sekvens, sorterade i icke avtagande ordning. Positionerna för elementen är $1, 2, 3, \dots, n$. Det finns också ett ytterligare element. Positionen för detta element i sekvensen ska bestämmas. Om samma element förekommer på flera positioner, så en av dessa positioner ska bestämmas.

Algoritm

Det sökta elementet jämförs först med det element som finns i mitten. Om de är likadana, så är positionen för det sökta elementet bestämd och algoritmen avslutas. I fall att det sökta elementet är mindre än det element som finns i mitten, så bestäms elementets position rekursivt i den del av sekvensen som finns framför mittersta elementet. I fall att det sökta elementet är större än det element som finns i mitten, så bestäms elementets position rekursivt i den del av sekvensen som finns efter mittersta elementet. I fall att inga delsekvenser återstår att genomsökas, så finns inte elementet i sekvensen.

Elementär operation

För att bestämma algoritmens tidskomplexitet, kan en jämförelse av det sökta elementet med ett element i sekvensen väljas som en elementär operation. En jämförelse ska omfatta alla de operationer som utförs för att bestämma om det sökta elementet är lika med, mindre än eller större än det element som finns i sekvensen.

Komplexitetsfunktionen ska ge antalet jämförelser för olika längder av sekvensen.

Tidskomplexiteten i värsta fall

(eng. worst-case time complexity)

Ett sätt att värsta fall inträffar är att det sökta elementet är större än alla element i sekvensen. Antalet jämförelser i det här fallet ska bestämmas.

Det sökta elementet jämförs med det element som finns i mitten av sekvensen. Det sökta elementet är större än detta element, så sökningen fortsätter i den delen av sekvensen som ligger efter mittersta elementet, om en sådan delsekvens finns. Det samma gäller för alla kommande jämförelser (på alla nivåer av rekursionen).

Om det finns $n > 1$ element i sekvensen, så finns det $\lfloor n/2 \rfloor$ element i den delsekvens som genomsöks. Därför gäller följande ekvation för antalet jämförelser:

$$W(n) = 1 + W\left(\left\lfloor \frac{n}{2} \right\rfloor\right)$$

I fall att det finns bara ett element i sekvensen, så jämförs bara det sökta elementet med detta element. Antalet jämförelser är i så fall 1. Alltså:

$$W(1) = 1$$

Antalet jämförelser i värsta fall kan beskrivas med följande differensekvation:

$$W(n) = 1 + W\left(\left\lfloor \frac{n}{2} \right\rfloor\right), n > 1$$

$$W(1) = 1$$

Lösningen till denna ekvation ger antalet jämförelser som binär sökning utför i värsta fall. Det här antalet är:

$$W(n) = \lfloor \log_2 n \rfloor + 1$$

$$W(n) \in \Theta(\log_2 n)$$