

TDDC91, TDDE22, 725G97 Lektion 1

Ordo-notation och tidskomplexitet

Magnus Nielsen
magnus.nielsen@liu.se

4 september 2018

Första 45

- Enkel for-loop med swap.
Givet programmet:

```
public void sortera(int[] l) {  
    for (int i = 0; i < l.length; i++) {  
        swap(l[0], l[i]);  
    }  
}
```

Låt studenterna fundera över, och diskutera i par eller några stycken, komplexiteten i några minuter. Samla ihop förslag, och visa sedan hur vi kan komma fram till att funktionen uppfyller:
 $O(n)$ (linjär komplexitet). ($\Theta(n)$).

Om studenterna ej sett en swap funktion förut (Även här kan det vara en god idé att visa att swap uppfyller $O(1)$ (konstant) ($\Theta(1)$):

```
public void swap(Integer a, Integer b) {  
    Integer c = a;  
    a = b;  
    b = c;  
}
```

- Nästlad for-loop.

Vi fortsätter bygga på det tidigare exemplet, mot att till slut ha en färdig bubble-sort. Sorteringen i sig blir ju fortfarande inte speciellt betydelsefull, men det är i nuläget irrelevant.

```
public void sortera(int[] l) {
    for (int i = 0; i < l.length; i++) {
        for (int j = l.length-1; j > 0 ; j--) {
            swap(l[j-1], l[j]);
        }
    }
}
```

Låt studenterna åter diskutera några minuter, och diskutera sedan åter i grupp.

$O(n^2)$ (kvadratisk). ($\Theta(n^2)$).

- Fungerande Bubble-sort.

Vi introducerar en if-sats för att göra vårt program till en fungerande sortering:

```
public void sortera(int[] l) {
    for (int i = 0; i < l.length; i++) {
        for (int j = l.length-1; j > 0 ; j--) {
            if (l[j] < l[j-1]) {
                swap(l[j-1], l[j]);
            }
        }
    }
}
```

I vanlig ordning diskussion. Hur ändras komplexiteten av if-satsen?

$O(n^2)$ (kvadratisk). ($\Theta(n^2)$).

- Optimerad Bubble-sort.

Vi optimerar Bubble-sort m.h.a. en boolean, vilket gör att exekveringen kan avbrytas redan efter en iteration av den inre loopen, samt att vi kortar den inre loopen (den går till i istället för ända ner till 0 då det inte finns behov av det). Hur påverkar detta tidskomplexiteten?

```
public void sortera(int[] l) {
    for (int i = 0; i < l.length; i++) {
        boolean swapped = false;
        for (int j = l.length-1; j > i ; j--) {
            if (l[j] < l[j-1]) {
                swapped = true;
                swap(l[j-1], l[j]);
            }
        }
        if (!swapped) {
            break;
        }
    }
}
```

Vi har fortfarande $O(n^2)$ (kvadratisk) komplexitet. Hur är det möjligt? (Det är en värsta-fallet analys).

I bästa fallet har vi: $\Omega(n)$, och därmed inget Theta.

Tryck gärna på att det är inte den exakta tiden som är intressant, namnet till trots, utan hur väl funktionen eller programmet skalar med storleken på indata.

Om mycket tid återstår kan det vara intressant att diskutera hur bästa-fallet och / eller amorterad analys (genomsnittlig körning) har förändrats genom stegen, annars borde det vara ungefär lagom att ta rast.

- Svar till lektionsuppgifterna.

1) Alla.

2) $O(n)$.

3) $O(n^3)$.

4) Hänvisa i första hand till reglerna som finns med på uppgiftsbladet.

Om $n \geq 1$ vet vi att $f(n) = 2n^3 + 3n + 18 \leq 2n^3 + 3n^3 + 18n^3 = 23n^3$.

Alltså: $f(n) \in O(n^3)$.

$f(n) = 2n^3 + 3n + 18 > n^3$ för alla $n \geq 1$.

Alltså: $f(n) \in \Omega(n^3)$.

Då funktionen uppfyller både O och Ω har vi att $f(n) \in \Theta(n^3)$.