Complexiteit

Uitwerkingen

Opgave 26

Het selectieprobleem (als gedefinieerd in sectie 6.1 van het dictaat) zoekt, voor een array getallen $A[1], A[2], \ldots, A[n]$ en een getal $1 \leq k \leq n$, het op $k-1$ na kleinste element. Laten we ons voor het gemak beperken tot $k = 1$, i.e. we zoeken het kleinste element uit de array. Stel we hebben alle elementen in de array vergeleken met een ander element in de array, op één element na. Zonder dit element met een ander element in de array te vergelijken hebben we geen informatie over dit element (we weten niet hoe het zich verhoudt tot de andere elementen) en weten we dus niet of dit element groter of kleiner is dan de elementen die we tot nu toe hebben bekeken, en zou het zomaar het element kunnen zijn wat we zoeken.

Als we elk paar van elementen met elkaar vergelijken als volgt hebben we (minstens) $\lceil \frac{n}{2} \rceil$ vergelijkingen nodig.

$$A[1] \leq A[2]? \qquad A[3] \leq A[4]? \qquad A[5] \leq A[6]? \qquad \ldots \qquad A[n-1] \leq A[n]?$$

Let op dat het vergelijken van elementen op deze manier nog niet het antwoord geeft. Er zijn dus nog meer vergelijkingen nodig om tot het antwoord te komen, maar we hebben *minstens* $\lceil \frac{n}{2} \rceil$ vergelijkingen nodig.

Opgave 30

(zie dictaat sectie 6.6)

```
[100, 80, 20, 70, 60, 33, 14, 15, 29, 17, 18, 44, 86, 50, 12]
```

1. Verdeel in groepjes van 5 elementen:

```
[100, 80, 20, 70, 60]
[33, 14, 15, 29, 17]
[18, 44, 86, 50, 12]
```

2. Vind mediaan van elk groepje:

```
[20, 60, 70, 80, 100], m1=70
[14, 15, 17, 29, 33],  m2=17
[12, 18, 44, 50, 86],  m3=44
```

3. Mediaan x van gevonden medianen:

```
[17, 44, 70], x=44
```

4. Partitioneer alle elementen rond x:

$\leq x$:`[20, 33, 14, 15, 29, 17, 18, 12, 44]`
$> x$:`[100, 80, 70, 60, 86, 50]`

5. $k = 5$ en het aantal getallen $\leq x$ is $m = 9$, dus we zoeken de k-de in grootte in de eerste partitie. Dit zoeken gebeurt recursief dus we roepen het algoritme aan op de partitie van elementen $\leq x$.

```
[20, 33, 14, 15, 29, 17, 18, 12, 44]
```

1. Verdeel in groepjes van 5 elementen:

```
[20, 33, 14, 15, 29]
[17, 18, 12]
```

2. Vind medianen van elk groepje:

```
[14, 15, 20, 29, 33] m1=20
[12, 17, 18]         m2=17
```

3. Mediaan x van gevonden medianen $x = 18.5$.

4. Partitioneer alle elementen rond x:

$\leq x$:`[14, 15, 17, 18, 12]`
$> x$:`[20, 33, 29, 44]`

5. $k = 5$ en het aantal getallen $\leq x$ is $m = 5$, dus we zoeken de k-de in grootte in de eerste partitie. De rij van 5 kunnen we niet nog verder in groepjes van 5 opsplitsen dus dit is een basis geval, en we kunnen (bijvoorbeeld door de rij te sorteren) het element wat 5-de in grootte is vinden.

```
18
```

Opgave 31

Loop lineair over de array heen, en als we twee nullen achter elkaar zien geven we true terug.

- Voor $n = 2$. Best case 1 vergelijking, worst case 2 vergelijkingen.

if $(A[1] == 0)$
    return $A[2] == 0$
else
    return $false$

- Voor $n = 3$. Best case 2 vergelijking (zou met 1 kunnen maar dit algoritme doet 2), worst case 2 vergelijkingen.

```
if (A[1] == 0)
    return A[2] == 0
else
    if (A[2] == 0)
        return A[3] == 0
    else
        return false
```

- Voor $n = 4$. Best case 2 vergelijkingen, worst case (dit algoritme) 4 vergelijkingen.

```
prev := (A[1] == 0)
for x := 2 to 4
    cur := (A[x] == 0)
    if (cur and prev)
        return true
    prev := cur
```

- Voor $n = 5$. Best case 2 vergelijkingen, worst case (dit algoritme) 5 vergelijkingen.

```
prev := (A[1] == 0)
for x := 2 to 5
    cur := (A[x] == 0)
    if (cur and prev)
        return true
    prev := cur
```

Voor bijvoorbeeld $n = 4$ kunnen we (ook in het slechtste geval) genoeg nemen met $3 < 4$ vergelijkingen. Het idee is dat we niet naar $A[1]$ hoeven te kijken als we al weten dat $A[2]$ gelijk is aan 1 (en idem dito voor $A[4]$ en $A[3]$).

```
if (A[2] == 0)
    if (A[3] == 0)
        return true
    else
        return A[1] == 0
else
    if(A[3] == 0)
        returnA[4] == 0
    else
        return false
```

Het geldt dus niet dat we voor n in het algemeen n vergelijkingen moeten doen in het slechtste geval (we hebben immers een tegenvoorbeeld), dus het zal onmogelijk zijn om een adversary argument te geven wat aantoont dat we in het slechtste geval n vergelijkingen moeten doen.

Opgave 36

Herhaaldelijk invullen (en de uitkomsten een beetje organiseren) geeft:

$$
\begin{aligned}
B(1) &= 0 \\
B(2) &= 0+1 && = 1 && = 1 \cdot 1 \\
B(4) &= 2 \cdot 1 + 2 && = 2+2 && = 2 \cdot 2 \\
B(8) &= 2 \cdot (2+2) + 4 && = 4+4+4 && = 3 \cdot 4 \\
B(16) &= 2 \cdot (4+4+4) + 8 && = 8+8+8+8 && = 4 \cdot 8 \\
B(32) &= 2 \cdot (8+8+8+8) + 16 && = 16+16+16+16+16 && = 5 \cdot 16 \\
&\vdots \\
B(n) &= 2 \cdot (\tfrac{n}{4} + \tfrac{n}{4} + \cdots + \tfrac{n}{4}) + \tfrac{n}{2} && = \tfrac{n}{2} + \tfrac{n}{2} + \tfrac{n}{2} + \cdots + \tfrac{n}{2} && = \log_2(n) \cdot \tfrac{n}{2}
\end{aligned}
$$

Als we niet overtuigd zijn door het patroon kunnen deze uitkomst ook bewijzen met inductie:

(i) $B(1) = \frac{1}{2} \cdot \log_2(1) = 0$ klopt.

(ii) Voor de inductie stap willen we, onder de aanname dat $B(n) = \frac{n}{2} \cdot \log_2(n)$, aantonen dat dezelfde vergelijking ook klopt voor $B(m)$ met $m = 2n$.

$$
\begin{aligned}
B(m) &= B(2n) \\
&= 2 \cdot B(\tfrac{2n}{2}) + \tfrac{2n}{2} && \text{(gegeven recursieve definitie)} \\
&= 2 \cdot B(n) + n \\
&= 2 \cdot \tfrac{n}{2} \cdot \log_2(n) + n && \text{(inductie aanname hierboven)} \\
&= \tfrac{m}{2} \cdot \log_2(\tfrac{m}{2}) + \tfrac{m}{2} \\
&= \tfrac{m}{2} \cdot \big(\log_2(\tfrac{1}{2} \cdot m) + 1\big) \\
&= \tfrac{m}{2} \cdot (-1 + \log_2(m) + 1) \\
&= \tfrac{m}{2} \cdot \log_2(m) && \square
\end{aligned}
$$

Opgave 49

We kiezen een willekeurige knoop v, en lopen door de graaf tot we weer terug komen bij v (we lopen alleen over takken die nog onbewandeld zijn, en bij meerdere mogelijkheden kiezen we de tak naar de knoop met de laagste index). Als we beginnen bij $v = 1$ levert dit de volgende kring op:

[1, 2, 3, 4, 5, 6, 7, 2, 8, 7, 9, 2, 13, 1]

Vervolgens kiezen we een knoop in deze kring met nog onbewandelde takken, zeg $v = 3$. Vanaf knoop 3 lopen we weer door de graaf (alleen over onbewandelde takken) en krijgen de volgende kring terug naar knoop 3:

[3, 10, 4, 9, 6, 12, 9, 11, 3]

Deze kring voegen we toe op de plek van knoop drie in de eerste kring en we krijgen:

$$[1, 2, 3, 10, 4, 9, 6, 12, 9, 11, 3, 4, 5, 6, 7, 2, 8, 7, 9, 2, 13, 1]$$

Nu zijn tevens alle takken van de graaf bezocht, dus het algoritme is klaar.