

11.30/31

Uitwerking tentamen Algoritmiek, donderdag 4 juni 2020

①

1).

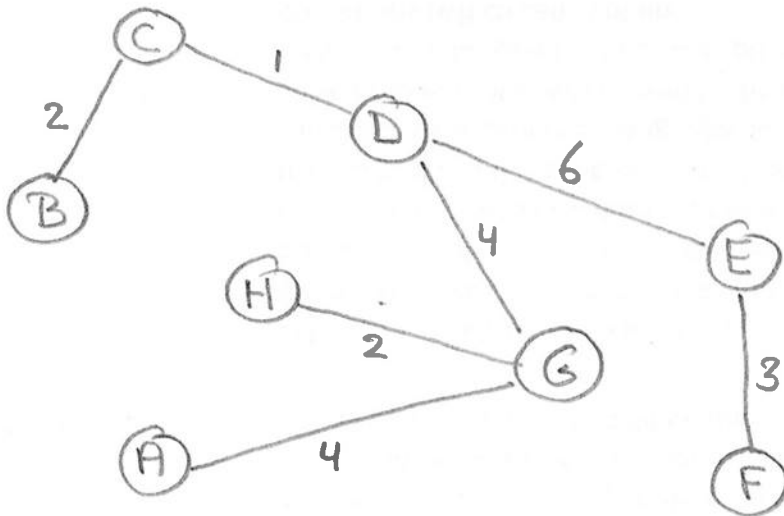
We lopen de takken in oplopende volgorde van gewicht af, en voegen ze toe aan een (initieel lege) takkenverzameling als ze daarmee geen kring vormen. We stoppen als we zeven takken hebben toegevoegd. (want er zijn acht knopen)

* tak $C \stackrel{1}{-} D$: voeg toe* tak $B \stackrel{2}{-} C$: voeg toe* tak $B \stackrel{2}{-} D$: voeg niet toe, want vormt kring* tak $G \stackrel{2}{-} H$: voeg toe* tak $E \stackrel{3}{-} F$: voeg toe* tak $A \stackrel{4}{-} G$: voeg toe* tak $D \stackrel{4}{-} G$: voeg toe* tak $A \stackrel{5}{-} H$: voeg niet toe, want vormt kring* tak $B \stackrel{5}{-} H$: voeg niet toe, want vormt kring* tak $D \stackrel{6}{-} E$: voeg toe

Stop, want zeven takken toegevoegd.

11.39

De resulterende minimale opspannende boom is



11.42

11.57

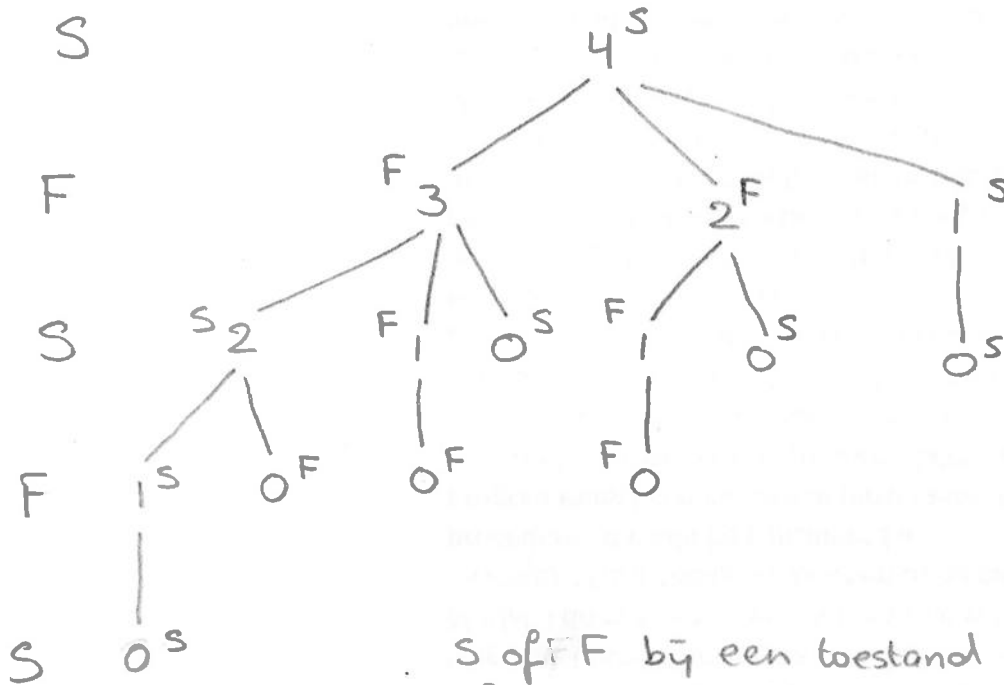
2(a)

Een toestand bestaat uit het aantal resterende lucifers i , met $0 \leq i \leq n$, en de speler die aan de beurt is.

Een toestand is een eindtoestand als $i=0$.

11.59 De acties zijn het weghalen van 1, 2 of 3 lucifers door de speler die aan de beurt is (gevolgd door het wisselen van beurt)

(b)
Speler aan beurt



S of FF bij een toestand geeft aan of de toestand winnend is voor S of F.

12.05/6

(c)

- $n=1 : 2$
- $n=2 : 4$
- $n=3 : 8$
- $n=4 : 15$

} rechtstreeks uit (b) gehaald.

Als $n=5$, kan beginspeler Suzan voor Freek een toestand met 4, 3 of 2 lucifers overlaten, door 1, 2 of 3 lucifers te pakken.

Dat geeft subbomen van de complete toestand-actie-boom met respectievelijk 15, 8 en 4 knopen.

Met de wortel van de boom erbij geeft dat:

$$1 + 15 + 8 + 4 = 28 \text{ knopen}$$

12.11

(d)

Allereerst geldt:

$$\left. \begin{array}{l} f(1) = 2 \\ f(2) = 4 \\ f(3) = 8 \end{array} \right\} \text{dit volgt uit (c)}$$

Vervolgens geldt voor $n \geq 4$:

$$f(n) = 1 + f(n-1) + f(n-2) + f(n-3)$$

De verklaring is net als bij (c) dat beginspeler Suzan voor $n \geq 4$, na haar eerste zet een toestand met $n-1$, $n-2$ of $n-3$ lucifers voor Freek kan overlaten. Dat geeft subbomen met $f(n-1)$, $f(n-2)$ en $f(n-3)$ knopen.

Met de wortel van de boom erbij geeft dat:

$$1 + f(n-1) + f(n-2) + f(n-3)$$

12.17/8/20

(e)

bool winnend (int stand)

{ int i; maxi;

if (stand == 0)

return true;

else // stand ≥ 1

{ maxi = min (stand, 3);

for (i=1; i \leq maxi; i++)

{ if (!winnend (stand - i))

return true;

}

return false;

}

}

12.28/9

(f) Een toestand met $i=1$ lucifer is verliezend voor de speler die aan de beurt is. Je moet dus proberen om de andere speler uiteindelijk een toestand te bezorgen met $i=1$ lucifer.

In iedere zet worden er 1, 2 of 3 lucifers weggehaald.

Als beide spelers een zet doen, kan de tweede speler er dus voor

altijd voor zorgen dat er bij elkaar 4 lucifers worden weggehaald.

Als $n \equiv 1 \pmod{4}$ en jy moet beginnen, dan kan de andere

speler er dus voor zorgen dat je uiteindelijk in een toestand

belandt met jou aan de beurt en nog $i=1$ lucifer, door herhaaldelijk samen 4 lucifers weg te halen.

Als $n \equiv 1 \pmod{u}$ is de toestand dus niet winnend voor de speler die aan de beurt is

Als $n \not\equiv 1 \pmod{u}$, dan kan de speler die aan de beurt is ervoor zorgen dat er een toestand met $i \equiv 1 \pmod{u}$ lucifers ontstaat, met de andere speler aan de beurt.

Dat is dus een winnende toestand voor de beginspeler.

in zijn eerst volgende zet

12.41 / 1257

3(a)

$P(i, j)$ stelt het aantal paden van lengte $i=0$ voor om van positie $(0, j)$ in positie $(0, k_2)$ te komen.

Als $j = k_2$, dan is dat aantal 1: het lege pad, want je bent al op je bestemming.

Als $j \neq k_2$, dan is dat aantal 0: je kunt niet in 0 stappen op een andere positie komen.

Als $i > 0$ is, dan moet je iedere stap naar een lagere r_y . Voor de eerste stap moet je dus naar r_y $i-1$. Vanuit kolom j kun je in die ene stap naar kolom $j-1$, kolom j of kolom $j+1$.

Vanaf daar moet je dan nog in $i-1$ stappen naar $(0, k_2)$ zien te komen. We tellen de mogelijke paden voor de drie opties bij elkaar.

Daarom geldt in principe:

$$P(i, j) = P(i-1, j-1) + P(i-1, j) + P(i-1, j+1)$$

Alleen als $j=0$, kun je niet naar kolom $j-1$, dus dan is

$$P(i, j) = P(i-1, j) + P(i-1, j+1) = P(i-1, 0) + P(i-1, 1)$$

En als $j=n$, kun je niet naar kolom $j+1$, dus dan is

$$P(i, j) = P(i-1, j-1) + P(i-1, j) = P(i-1, n-1) + P(i-1, n)$$

13.08/9/10

13.10

(b)

int paden (int n, int k1, int k2)

{ int P[n+1][n+1],
 i, j;

 for (j=0; j<=n; j++)

 if (j==k2)

 P[0][j]=1;

 else

 P[0][j]=0;

 for (i=1; i<=n; i++)

 P[i][0] = P[i-1][0] + P[i-1][1];

 for (j=1; j<n; j++)

 P[i][j] = P[i-1][j-1] + P[i-1][j] + P[i-1][j+1];

 P[i][n] = P[i-1][n-1] + P[i-1][n];

 }

 return P[n][k1];

}

13.19/20

(c)

| | j=0 | 1 | 2 | 3 | 4 |
|-----|-----|----|----|----|---|
| i=4 | 12 | 18 | 16 | 10 | 4 |
| 3 | 5 | 7 | 6 | 3 | 1 |
| 2 | 2 | 3 | 2 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |

Het eindantwoord is 16.

13.23

(d) Een basisoperatie is de toekenning

$$P[i][j] = P[i-1][j-1] + P[i-1][j] + P[i-1][j+1];$$

In de binnenste for-lus wordt die operatie $n-1$ keer uitgevoerd. De buitenste for-lus kent n iteraties, dus in totaal wordt de basis operatie $n \times (n-1)$ keer uitgevoerd.

Dat is in $\Theta(n^2)$, dus dat is de tijdscomplexiteit.

13.27

4 (a)

```

bool aanval (int n, pair<int,int> dame1, pair<int,int> dame2,
             pair<int,int> pion
)
{
    if (dame1.first == dame2.first) // zelfde rij
        if (pion.first != dame1.first) // zeker niet ertussen
            return true;
        else // ook zelfde rij
            if (pion.second < min (dame1.second, dame2.second)
                || pion.second > max (dame1.second, dame2.second))
                return true; // niet ertussen
            else
                return false;

    if (dame1.second == dame2.second) // zelfde kolom
        if (pion.second != dame1.second)
            return true;
        else // ook zelfde kolom
            if (pion.first < min (dame1.first, dame2.first)
                || pion.first > max (dame1.first, dame2.first))
                return true;
            else
                return false;

    if (dame1.second - dame2.second ==
        dame1.first - dame2.first) // zelfde diagonaal
        if (dame1.second - pion.second !=
            dame1.first - pion.first)
            return true
        else // ook zelfde diagonaal
            if (pion.first < min (dame1.first, dame2.first)
                || pion.first > max (dame1.first, dame2.first))
                return true;
            else
                return false;
}

```

13.45 / 16.38

```

if (dame1.second - dame2.second ==
    dame2.first - dame1.first) // zelfde diagonaal
    if (dame1.second - pion.second !=
        pion.first - dame1.first)
        return true
    else // ook zelfde diagonaal
        if (pion.first < min (dame1.first, dame2.first)
            || pion.first > max (dame1.first, dame2.first))
            return true;
        else
            return false;
}

```

// als je hier komt, staan de twee dames in verschillende
 // rjen, kolommen en diagonalen.
 return false;

}

16.45

(b)

int plaatsdames (int n, vector<pair<int,int>> dames,
 pair<int,int> pion)

{ int k, aantal, maxrj, rj, kolom
 pair<int,int> pos;

k = dames.size();

if (k == n) // complete plaatsing

return 1;

else

{ aantal = 0; // nog 0 plaatsingen

// we gaan de k+1 'e dame plaatsen

// die mag maximaal in rj k+2 komen

maxrj = min(n, k+2);

if (k == 0)

{ rj = 1;

kolom = 1;

}

else

16.54/8

{ rj = dames[k-1].first;

// we gaan verder na de vorige

kolom = dames[k-1].second + 1;

// dame, om dubbeltellen te

}

// voorkomen.

while (rj ≤ maxrj)

// we kunnen deze positie nog proberen

{

// voor deze dame

if (kolom > n)

{ rj ++;

kolom = 1;

}

else // geldige positie.

{ pos = make_pair(rj, kolom);

if (pos != pion)

{ if (!geeftaanval(n, dames, pion, pos)) // het mag!

{ dames.push_back(pos);

aantal += plaatsdames(n, dames, pion);

dames.pop_back();

}

}

17.22

```
        kolom++;  
    } // else, geldige positie  
} // while.  
return aantal;  
} // else, nog geen complete plaatsing  
  
} // plaatsdames.
```

17.25