

## Werkcollege 6

### Backtracking, divide and conquer, decrease and conquer en partitionering

Jan van Rijn

Leiden Institute of Advanced Computer Science

17 maart 2011

## Traveling Salesman Problem

Gegeven een complete, ongerichte graaf met  $n$  knopen en met gewichten op de takken. Geef de Hamiltonkring met minimaal totaal gewicht.

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

a

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

$$a \xrightarrow{10} b$$

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

$$a \xrightarrow{10} b \xrightarrow{19} c$$

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

$$a \xrightarrow{10} \quad b \xrightarrow{19} \quad c \xrightarrow{46} \quad d$$

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

$$a \xrightarrow{10} \quad b \xrightarrow{19} \quad c \xrightarrow{46} \quad d \xrightarrow{58} \quad e$$

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

$$a \xrightarrow{10} b \xrightarrow{19} c \xrightarrow{46} d \xrightarrow{58} e \xrightarrow{64} a$$



# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

a  $\xrightarrow{10}$  b  $\xrightarrow{19}$  c  $\xrightarrow{46}$  d  $\xrightarrow{58}$  e  $\xrightarrow{64}$  a  
a

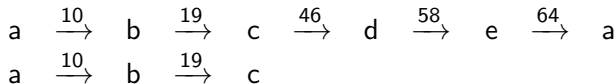
# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

$$\begin{array}{ccccccccc} a & \xrightarrow{10} & b & \xrightarrow{19} & c & \xrightarrow{46} & d & \xrightarrow{58} & e & \xrightarrow{64} & a \\ a & \xrightarrow{10} & b & & & & & & & & \end{array}$$

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)



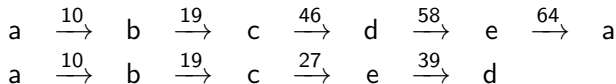
# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

$$\begin{array}{cccccc} a & \xrightarrow{10} & b & \xrightarrow{19} & c & \xrightarrow{46} & d & \xrightarrow{58} & e & \xrightarrow{64} & a \\ a & \xrightarrow{10} & b & \xrightarrow{19} & c & \xrightarrow{27} & e & & & & \end{array}$$

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)



# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

$$\begin{array}{cccccc} a & \xrightarrow{10} & b & \xrightarrow{19} & c & \xrightarrow{46} & d & \xrightarrow{58} & e & \xrightarrow{64} & a \\ a & \xrightarrow{10} & b & \xrightarrow{19} & c & \xrightarrow{27} & e & \xrightarrow{39} & d & \xrightarrow{50} & a \end{array}$$

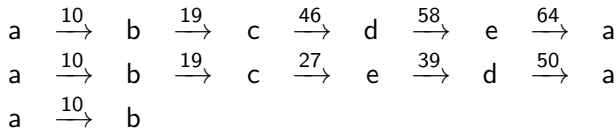
# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

a	$\xrightarrow{10}$	b	$\xrightarrow{19}$	c	$\xrightarrow{46}$	d	$\xrightarrow{58}$	e	$\xrightarrow{64}$	a
a	$\xrightarrow{10}$	b	$\xrightarrow{19}$	c	$\xrightarrow{27}$	e	$\xrightarrow{39}$	d	$\xrightarrow{50}$	a
a										

# Opgave 1

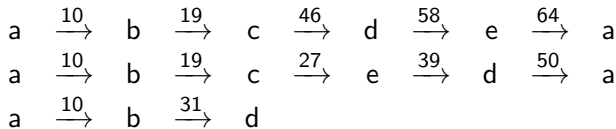
Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)





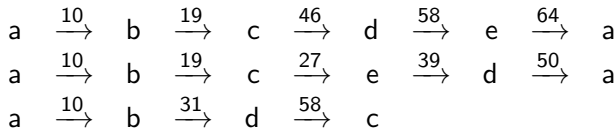
# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)



# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)



# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

a  $\xrightarrow{10}$  b  $\xrightarrow{19}$  c  $\xrightarrow{46}$  d  $\xrightarrow{58}$  e  $\xrightarrow{64}$  a  
a  $\xrightarrow{10}$  b  $\xrightarrow{19}$  c  $\xrightarrow{27}$  e  $\xrightarrow{39}$  d  $\xrightarrow{50}$  a  
a  $\xrightarrow{10}$  b  $\xrightarrow{31}$  d  $\xrightarrow{58}$  c  
...  
a

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

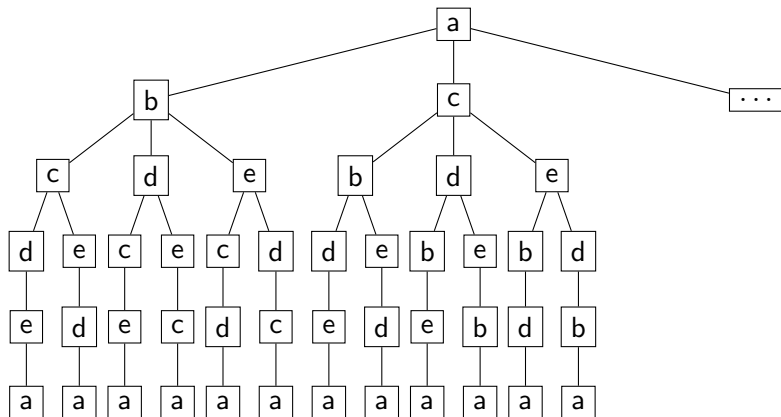
a  $\xrightarrow{10}$  b  $\xrightarrow{19}$  c  $\xrightarrow{46}$  d  $\xrightarrow{58}$  e  $\xrightarrow{64}$  a  
a  $\xrightarrow{10}$  b  $\xrightarrow{19}$  c  $\xrightarrow{27}$  e  $\xrightarrow{39}$  d  $\xrightarrow{50}$  a  
a  $\xrightarrow{10}$  b  $\xrightarrow{31}$  d  $\xrightarrow{58}$  c  
...  
a  $\xrightarrow{24}$  c

# Opgave 1

Genereer alle permutaties (e.g. *abdce*, let op dat je ook nog terug naar *a* moet.)

a	$\xrightarrow{10}$	b	$\xrightarrow{19}$	c	$\xrightarrow{46}$	d	$\xrightarrow{58}$	e	$\xrightarrow{64}$	a
a	$\xrightarrow{10}$	b	$\xrightarrow{19}$	c	$\xrightarrow{27}$	e	$\xrightarrow{39}$	d	$\xrightarrow{50}$	a
a	$\xrightarrow{10}$	b	$\xrightarrow{31}$	d	$\xrightarrow{58}$	c				
...										
a	$\xrightarrow{24}$	c	$\xrightarrow{51}$	d						

# Opgave 1



## Opgave 2

```
Algorithm maxIndex( array[0 ... n-1] ){
  if( n-1 == 0 ) return links;
  else {
    int linkerdeel = maxIndex( A[0 ... floor(n-1)/2 ] );
    int rechterdeel = maxIndex( A[floor(n-1)/2 ... n-1] );

    if( A[linkerdeel] < A[rechterdeel] ){
      return rechterdeel;
    } else {
      return linkerdeel;
    }
  }
}
```

Waarbij `floor(getal)` een decimaal als parameter accepteert, en deze naar beneden afrondt.

# Opgave 7

```
Algorithm negativeBeforeOthers( A[0 ... n-1] ) {
    int i = 0;
    int j = n-1;
    while( i <= j ) {
        if( A[i] < 0 ){
            i++;
        } else {
            verwissel( A[i], A[j] );
            j--;
        }
    }
}
```

(Zie sheets hoorcollege)



# Opgave 8

```
Algorithm dutchFlag( A[0 ... n-1] ){
  int r = 0;
  int w = 0;
  int b = n-1;
  while( w <= b ){
    if( A[w] == 'R' ){
      verwissel( A[r], A[w] );
      r++;
      w++;
    } else if( A[w] == 'W' ) {
      w++;
    } else {
      verwissel( A[w], A[b] );
      b--;
    }
  }
}
```