

# Algorithms and Datastructures assignment 3

Thomas Broby Nielsen (xlq119)

Tobias Overgaard (vqg954)

Christian Buchter (zvc154)

1. juni 2015

## Indhold

<b>1</b>	<b>Task 1</b>	<b>2</b>
<b>2</b>	<b>task 2</b>	<b>3</b>
<b>3</b>	<b>Task 3</b>	<b>4</b>

# 1 Task 1

```
possible(p, b, n)
1 r=0
2 for m=1 to p.length
3     if m=p.length
4         if r >= 0 return true
5         else return false
6     if b[m]+r > n+((p[m+1]-p[m])*2)
7         r=r+(b[m]-(n+((p[m+1]-p[m])*2)))
8     if b[m]+r < n
9         r=r-(n-(b[m]+(p[m+1]-p[m])*2))
10    else r=r+0
```

## 2 task 2

Vores algoritme har Greedy choice Property.

Det grådige valg går ud på at den første bar skal have præcist  $\bar{b}$  øl. Da distancen mellem barerne er linær, koster det ikke ekstra øl at stoppe på alle barerne. Ved den første bar er der to valgmuligheder. Hvis  $b_1 < \bar{b}$  så skal der sendes  $\bar{b} - b_1 + (p_2 - p_1) \cdot 2$  øl til bar 1 fra bar 2. Hvis  $b_1 > \bar{b}$  så kan den optimale løsning ikke sende mere end  $b_1 - \bar{b}$ . Det grådige valg virker fordi at hvis Der skal kunne være mindst  $\bar{b}$  øl på hver bar, skal der være mindst  $\bar{b}$  øl på denne bar. Dette gælder for alle barrierne. Derfor kan vi se på barrierne fra en ende af og løse problemet et subproblem af gangen. og derefter kigge på resten

Problemet går ud på at finde ud af om det er muligt at få  $\bar{b}$  øl i hver bar. Dette kan løses ved at finde ud af om det er muligt at have nok øl i en bar + resten. Den grådige løsning løser dette problem på en bar. Dette udviser optimal delstruktur da det grådige valg udføres på hver bar, hvilket giver en optimal løsning for det originale problem.

### 3 Task 3

```
maxBeer(p, b)
1 l=0
2 h=max(b)
2 while l < h
3     m = ceiling((l+h)/2)
4     if possible(b,p,m)
5         l=m
6     else h=m-1
7 return h
```

Denne algoritme vil bruge  $\log B$  iterationer af possible, hvilket giver køretiden:  $O(n \lg B)$ .

Vores algoritme har Greedy choice property, da den tjekker om det  $\text{ceiling}((l+h)/2)$  er løsningen. Hvis dette ikke er korrekt ændres  $l$  til  $m$  eller  $h$  til  $h-1$ . Dette virker da hvis possible giver "true" så er  $m$  muligt og  $l=m$ . Hvis ikke det er muligt prøves igen med et  $h=m-1$ . Det grådige valg er rigtigt fordi at hvis dette gentages nok vil det mindste tal ende med at være lig det største tal og det vil så være det maksimale tal som det er muligt for barrierne at have.

Problemet går ud på at finde det største mulige tal som alle barrierne kan have af øl. Dette tjekkes ved at bruge possible på et tal, indtil det korrekte tal er fundet. Derfor har problemet en optimal delstruktur.