

Algorithms and Datastructures assignment 3

Thomas Broby Nielsen (xlq119)

Tobias Overgaard (vqg954)

Christian Buchter (zvc154)

21. maj 2015

Indhold

1	Task 1	2
2	task 2	3
3	Task 3	4

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] > n+((m+1)-m)
7         r=r+(b[m]-(n+((m+1)-m)))
8     if b[m]< n
9         r=r-(n-(b[m]+((m+1)-m)))
10    else r=r+0
```

2 task 2

Problemet har optimal substruktur da i kun skal finde possibility, så er hvilken som helst løsning hvor hver bar får \bar{b} øl en optimal løsning.

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 $\bar{b} - b_1 + (p_2 - p_1)$ øl til bar 1 fra bar 2. Hvis $b_1 > \bar{b}$ så kan den optimale løsning ikke sende mere end $b_1 - \bar{b}$.

3 Task 3

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

Denne algoritme vil bruge $\log B$ iterationer af maxBeer, hvilket giver køretiden: $O(n \lg B)$ Algoritmen virker fordi at hvis vi kan få x øl i hver bar, kan vi også få $x' < x$ øl ved først at få øl, og så bare kassere øl ved hver bar.