Assignment 3 - AlgDat

Simon Warg BCS315

20. maj 2014

Task 1

Algorithm 1 CanDistribute

---

for $i = 1$ to $n - 1$ **do**
  $distance = p_{i+1} - p_i$
  if $b_i < \hat{b}$ **then**
    $b_{i+1} = b_{i+1} - (\hat{b} - b_i) - 2 * distance$
    $b_i = \hat{b}$
  end if
  if $b_i > \hat{b}$ **then**
    $b_{i+1} = max(0, b_i - \hat{b} - 2 * distance)$
    $b_i = \hat{b}$
  end if
end for
return $b_n \geq \hat{b}$

Task 2

Hvis $b_1 < \hat{b}$, så vil den billigste, og dermed den beste, løsningen være at tage øl fra b_2. Hvis b_2 derefter får mindre end $\hat{b}$ øl så vil den endeste måde være at tage øl fra b_3. Det andet case; hvis $b_1 > \hat{b}$, så vil den beste løsningen være at sørge for at b_2 får så mange øl fra b_1 som mulig indtil $b_1 = \hat{b}$. Dermed gælder at for hver b_i så er $b_1 ... b_{i-1} \leq \hat{b}$. Hvis den sidste bar $b_n \geq \hat{b}$ ved vi at alle barer kan have $\hat{b}$ øl.

Task 3

Algorithm 2 Find maximum number of beer that can be distributed among all bars

$left = 0$
$right = B$
while $right - left > 1$ **do**
  if $CanDistribute(\lfloor\frac{right+left}{2}\rfloor)$ **then**
    $left = \lfloor\frac{right+left}{2}\rfloor$
  else
    $right = \lfloor\frac{right+left}{2}\rfloor$
  end if
end while
if $CanDistribute(right)$ **then**
  return right
else
  return left
end if

Funktionen kører i $O(\log B) + O(n)$ fordi i hvert iteration så tjekkes det hvis maximum m, $0 < m < B$, i et søgområde der halveres efter hver iteration. Hver gang den søger i søgområdet bruger den $O(n)$ tid, altså kører funktion i $O(n \log B)$.
Funktionen er korrekt siden den hver gang tjekker hvis maximum er under eller over $B/2$. Er den under så fortsætter den at undersøge om maximum er under eller over $B/2 - B/4$, er den over så tjekker den hvis maximum er over eller under $B/2 + B/4$ indtil søgområdet er kun to tal, hvor den tjekker begge for maximum.