Karol Cidyło

Zadanie 2 z listy 3.

Algorytm

```
Procedure MaxMin(S:set)
if |S|= 1 then return {a1, a1}
else
    if |S|= 2 then return (max(a1, a2), min(a1, a2))
    else
        podziel S na dwa rownoliczne (z dokladnoscia do jednego elementu)
        podzbiory S1, S2
        (max1, min1) <- MaxMin(S1)
        (max2, min2) <- MaxMin(S2)
        return (max(max1, max2), min(min1, min2))
```

.

$T(n) = T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + 2$

Lemat
Weźmy $n = 2^i + k$, gdzie $k < 2^i$:

$$T(n) = \begin{cases} \frac{3}{2}2^i - 2 + 2k & \text{jeśli } k \le 2^{i-1} \\ 2 \cdot 2^i - 2 + k & \text{jeśli } k > 2^{i-1} \end{cases}$$

Dowód indukcyjny po i
Mając $n = 2^i + k$, gdzie $k < 2^i$:

1. **Podstawa:**
   Dla i = 1
   (A) k = 0. $T(2^1 + 0) = T(2) = 3 \cdot 1 + 2 \cdot 0 - 2 = 1$
   (B) k = 1. $T(2^1 + 0) = T(2) = 3 \cdot 1 + 2 \cdot 1 - 2 = 3$
2. **Założenie:**
   Załóżmy, że dla i - 1 jest to spełnione, czyli dla $\frac{n}{2} = \frac{2^{i-1}+k}{2}$
3. **Krok indukcyjny:**
   (A) $k \le 2^{i-1}$
   $\frac{n}{2} = \frac{2^{i-1}+k}{2}$
   $\lceil \frac{n}{2} \rceil = 2^{i-1} + \lceil \frac{k}{2} \rceil$
   $\lceil \frac{k}{2} \rceil \le 2^{i-2}$, $\lfloor \frac{k}{2} \rfloor \le 2^{i-2}$
   $T(n) = T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + 2 =$
   $= \frac{3}{2}2^{i-1} - 2 + 2\lfloor \frac{k}{2} \rfloor + \frac{3}{2}2^{i-1} - 2 + 2\lceil \frac{k}{2} \rceil + 2 = \frac{3}{2}2^i - 2 + 2k$

(B) $k > 2^{i-1}$
$2^{i-1} < \lceil \frac{k}{2} \rceil < 2^{i+1}$, $2^{i-1} < \lfloor \frac{k}{2} \rfloor < 2^{i+1}$
$T(n) = T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + 2 = T(\lfloor \frac{2^i+k}{2} \rfloor) + T(\lceil \frac{2^i+k}{2} \rceil) + 2 =$
$= 4 \cdot 2^{i-2} + \lfloor \frac{k}{2} \rfloor - 2 + 4 \cdot 2^{i-2} + \lceil \frac{k}{2} \rceil - 2 + 2 = 2 \cdot 2 \cdot (2^{i-2} + 2^{i-2}) + k - 2 =$
$= 2 \cdot 2^i - 2 + k$

□

Oznaczmy przez G(n) funkcję, która wyznacza różnicę w liczbie porównań. W takim razie:

$$G(n) = T(n) - \lceil \frac{3}{2} n - 2 \rceil$$

Podstawiając $n = 2^i + k$ otrzymujemy:

$$G(n) = T(n) - \lceil \frac{3}{2} 2^i + \frac{3}{2} k - 2 \rceil = T(n) - \frac{3}{2} 2^i + 2 - \lceil \frac{3}{2} k \rceil$$

Wykorzystując $T(n)$ z lematu mamy:

$$G(n) = \begin{cases} \lfloor \frac{k}{2} \rfloor & \text{jeśli } k \le 2^{i-1} \\ 2^{i-1} - \lceil \frac{k}{2} \rceil & \text{jeśli } k > 2^{i-1} \end{cases}$$

- Wartości dla, których algorytm wykonuje $\lceil \frac{3}{2} n - 2 \rceil$ porównań:
  To są te wartości k, dla których $G(n) == 0$, czyli 0, 1, $2^i - 1$.

- Maksymalna różnica ($G(n)$) jest wtedy, gdy $k = 2^{i-1}$ czyli: $G(n) = 2^{i-2}$
  $n = 2^i + k = 2^i + 2^{i-1}$
  $G(n) = \frac{n}{6}$

- ```
  Procedure MaxMin(S:set)
  if |S|= 1 then return {a1, a1}
  else
      if |S|= 2 then return (max(a1, a2), min(a1, a2))
      if |S| mod 2 == 0 and |S|/2 mod == 1
          podziel S na dwa zbiory o liczbie elementow:
          |S|/2 + 1 oraz |S|/2 - 1
      else
          podziel S na dwa rownoliczne
          (z dokladnoscia do jednego elementu)
          podzbiory S1, S2
      (max1, min1) <- MaxMin(S1)
      (max2, min2) <- MaxMin(S2)
      return (max(max1, max2), min(min1, min2))
  ```
  ```