Augmented data structures

Given a communication channel, process a sequence of requests.

Let a request be represented as a tuple $<x, t_1, t_2>$ where $x (0 < x \leq 1)$ is a portion of channel from $t_1$ to $t_2$. Let $\alpha, \beta, \gamma$.

**Algorithm**

If portion $x$ is available from $t_1$ to $t_2$, then schedule the request.
Else respond with "no".

Example

$\theta(0.2, 0, 18)$

Request $(0.4, 7, 15)$

Solution Structure

$O(\alpha + \beta + \gamma)$

Time

$O(n)$
2nd solution structure
2-3-4 trees

⇒ leaves hold nonoverlapping time intervals

* step on Augmented DS
1. Choose an underlying DS
2. Determine additional information to maintain in the underlying DS
3. Verify that we can maintain the additional info for the basic modified operations the under DS
4. Develop new operations

Idea: add info for χ to certain intermediate interval nodes that cover the req. interval!

⇒ add to each node χ

Field

\[ D(u) : \delta \text{ between } (\Delta) \text{ change} \]
\[ D(u) : \Delta \text{ (change)} \]

\[ M(u) : \text{ Max amount of capacity used at any time for } u \text{'s interval} \]

- leaves \([t', t'']\)
  1) true portion of channel in use during time \([t', t'']\) is
  \[
  \sum_{u \in P} D(u)
  \]
  \(P\) is path from root to the leaf

2) \[ M(u) = \max \sum_{u \in P} D(u) \]
Where \(P\) is any path from \(u\) down to leaf.

- test request processing

(c, \(t_1\), \(t_2\))

1) Find leaf \(l_1\) that can contain \(t_1\) in it's interval

2) Split tree into \(T_1, l_1, T_0\) as follows

For each node \(v \neq l_1\) on the path from root to \(l_1\) in order starting from root.

Do for each child \(w\) of \(v \neq l_1\) do
\begin{algorithm}
\begin{algorithmic}
  \STATE \textbf{for} each child \( u \) of \( v \) \textbf{do}
  \STATE \( D(c(u)) \leftarrow D(u) + D(c(v)) \)
  \STATE \( M(c(u)) \leftarrow M(u) + D(c(v)) \)
  \STATE \textbf{end for}
  \STATE \( D(c(v)) \leftarrow 0 \)
  \STATE \textbf{end for}
\end{algorithmic}
\end{algorithm}

Perform the split, updating the \( M \) values as segments are merged.

3) union \( l_1 \) to \( T_0 \) to get \( T'_0 \)

4) find the right most leaf \( l_2 \) in \( T'_0 \) containing the time \( \ell_2 \)

5) similar to 2) split \( T'_0 \) to \( T_2, l_2, T_3 \)

6) union the \( l_2 \) and \( T_3 \) to get \( T'_2 \)

\[ P(r) = 0.1 \]
\[ M(r) = \begin{cases} 0 & \text{if } M(\text{root of } T_i') \text{ is channel available} \\ \ell_2 & \text{otherwise} \end{cases} \]
\[ T_1' \leftarrow \text{let } y = M\text{ (root of } T_1') \]

\( c \leq 1 - y \)

\[ \text{then request can be processed} \]

\[ \text{else } \] cannot be processed. \[ \text{if request cannot be processed} \]

merge \( T_1, T_2', T_3 \)

do if req. satisfied do

- from \( l_1 \) with interval \( [t_1', t_2'] \)

- from \( l_2 \) do likewise

\[ [t_2', t_2''] \rightarrow [t_2', t_2], [t_2', t''] \]

request, \( c > c, t_1, t_2 \)

- add \( x \) to \( D\text{ (root of } T_2'') \)

- add \( x \) to \( M\text{ (root of } T_2'') \)

- merge \( T_1, l_1', T_2', l_2', T_3 \)

- Take care of the \( M \) values.

\[ \text{Case} \]

\[ D_{113} = 0\]

\[ \text{test } (0.2, 0, 18) \]
\[
\begin{align*}
R(4) &= 0.2 \
M(2) &= 0.5 \
D(5) &= 0.3 \
M(6) &= 0.2 \
D(7) &= 0 \\
M(7) &= 0.1 \\
\end{align*}
\]

\[
\text{req} : (0.1, 7, 15)
\]

\[
\begin{align*}
D(1) + D(2) + D(4) &= 0.2 \
D(1) + D(2) + D(5) &= 0.5 \
D(1) + D(3) + D(6) &= 0.2 \
D(1) + D(3) + D(7) &= 0
\end{align*}
\]

\[
\text{def } \mathcal{D} \\
\text{def } \mathcal{M} \\
MC(U) = \mathcal{D}(U) + \text{Max } \{ \mathcal{M}(U) \}
\]

\(u \text{ child of } \mathcal{S}\)