http://courses.csail.mit.edu/6.006

Administrivia

Course overview

"Peak finding" problem
- 1D version
- 2D version

Course Overview

- Efficient procedures for solving problems on large inputs (e.g., US highway map, human genome)

- Scalability

- Classic data structures and elementary algorithms (CLRS text)

- Real implementations in Python

- Fun problem sets
Content

8 modules each with motivating problem and problem set(s) (except last)

Algorithmic thinking: Peak finding

Sorting & Trees: Event simulation

Hashing: Genome comparison

Numerics: RSA encryption

Graphs: Rubik's cube

Shortest Paths: Caltech → MIT

Dynamic Programming: Image compression

Advanced Topics
PEAK FINDER

One-dimensional version

\[ a, b, c, d, e, f, g, h, i \]

\[ 1, 2, 3, 4, 5, 6, 7, 8, 9 \]

\( a \)-\( i \) are numbers

Position 2 is a peak if and only if
\( b \geq a \) and \( b \geq c \)

Position 9 is a peak if \( i \geq h \)

Problem: Find a peak if it exists.

* Does it always exist?

STRAIGHTFORWARD ALGORITHM

Start from left

\[ 1, 2, \ldots, n/2, \ldots, n-1, n \]

\[ \uparrow \]

might be peak

\[ \Theta(n) \] complexity worst case

What if we start in the middle?

Look at \( n/2 \) elements

Could look at \( n \) elements

Look at \( n/2 \) elements
Can we do better?

1 2 ... \( n/2-1 \) \( n/2 \) \( n/2+1 \) ... \( n-1 \) \( n \)

Look at \( n/2 \) position

If \( a[n/2] < a[n/2-1] \) then only look at left half \( 1 \ldots n/2-1 \) to look for peak.

Else if \( a[n/2] < a[n/2+1] \) then only look at right half \( n/2+1 \ldots n \) to look for peak.

Else \( n/2 \) position is a peak.

**Why?**

\[ a[n/2] \neq a[n/2-1] \]
\[ a[n/2] \neq a[n/2+1] \]

What is the complexity?

\[ T(n) = \sum_{i=1}^{\log_2 n} \Theta(1) \]

\[ T(n) = \Theta(\log_2 n) \]

\( n = 1,000,000 \)

\( \Theta(n) \) algo 13.5 in python impl

\( \Theta(\log n) \) algo 0.001 s

Argue that the algorithm is correct.

* In order to sum up the \( \Theta(1) \)'s as we do here, we need to find a constant that works for all.
2-Dimensional Version

$\begin{array}{cccc}
& c & b & d & e \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
14 & 13 & 12 & & 15 & 19 & 11 & 17 \\
16 & 17 & 19 & 20 & & & & \\
\end{array}$

$a$ is 2D peak iff $a > b$, $a > d$, $a > c$, $a > e$

Greedy ascent algorithm: $\Theta(nm)$ complexity

$\Theta(n^2)$ algorithm if $m = n$

0 peak

Extend 1D divide & conquer to 2D: Attempt #1

Pick middle column $j = \lfloor m/2 \rfloor$

Find a 1D peak at $i, j$

Use $(i, j)$ as a start point on row $i$ to find 1D-peak on row $i$
**ATTEMPT #1 FAILS**

Problem: 2D peak may not exist on row i and end up with 14 which is not a 2D peak.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>13</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

**ATTEMPT #2**

Pick middle column $j = n/2$

Find global maximum on column $j$ at $(i, j)$

Compare $(i, j-1), (i, j), (i, j+1)$

Pick left cols if $(i, j-1) > (i, j)$ (similarly for right)

$(i, j)$ is a 2D-peak if neither condition holds

Solve the new problem with half the number of columns

When you have a single column, find global maximum and you're done.
**Example of Attempt #2**

1. Pick this column: 10 8 10 10
   14 13 12 11
   15 9 11 21
   16 17 19 20

2. Global maximum for column 17 goes with:
   10 10
   12 11
   11 21
   19 20

3. Pick this column: 19 is global maximum for column 21.

**Complexity of Attempt #2**

- $h$ rows, $m$ columns
- $T(n, m) = T(n, m/2) + \Theta(n)$

To find global maximum on a column ($n$ rows):

$$T(n, m) = \Theta(n) + \ldots \Theta(n) \over \log m = \Theta(n \log m) = \Theta(n \log n)$$

Q: What if we replaced global maximum with 10-peak in Attempt #2? Would that work?