please don't grade

CSE 548 – Analysis of Algorithms, Spring 2013

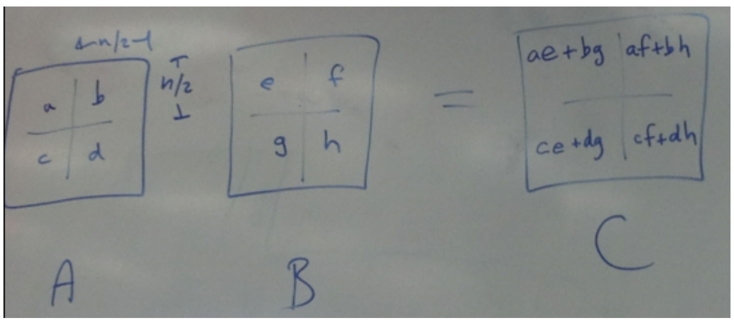
Assignment #2a

Duo Xu (#108662210)

partner: Yu-Yao Lin (#109090793)

Problem 1

(a) In the class, we had the following divide and conquer algorithm,



This algorithm is $O(n^3)$ time. The recurrence relation is $T(n) = 8T(n/2) + n^2$

Now the two matrices are in disk. Using DAM model, the $N * N$ matrix becomes $\frac{N}{\sqrt{B}} * \frac{N}{\sqrt{B}}$ blocks. Then using the above algorithm, the difference is that we treat it as a new $\frac{N}{\sqrt{B}} * \frac{N}{\sqrt{B}}$ matrix, each element in this matrix is a block instead of a number. Thus compared to the above algorithm, the recurrence relation is

$T(N) = 8T(N/2) + \frac{N^2}{B}$.

$T(N) = \frac{N^2}{B}$ when $N \leq \sqrt{B}$

Thus we need $O((\frac{N}{\sqrt{B}})^3) = O(N^3/B^{3/2})$ memory transfers for a multiplication.

(b) The difference from (a) is the size of each element of the matrix, (a) is $\frac{N}{\sqrt{B}} * \frac{N}{\sqrt{B}}$, while (b) is $\frac{N}{\sqrt{M}} * \frac{N}{\sqrt{M}}$. Thus the recurrence relation is

$T(N) = 8T(N/2) + \frac{N^2}{B}$

$T(N) = \frac{N^2}{B}$ when $N \leq \sqrt{M}$

In every recursion, the total memory transfers is the same as (a). However, as the size of each element is different, the total number of recursion is changed.

Let $N = 2^l$ and $2^k = \sqrt{M}$, thus $2^{l-k} = \frac{N}{\sqrt{M}}$

$$\begin{aligned}
T(2^l) &= 8T(2^{l-1}) + \tfrac{1}{B} * 2^{2l} \\
&= 8(8T(2^{l-2}) + \tfrac{1}{B} * 2^{2(l-1)}) + \tfrac{1}{B} * 2^{2l} \\
&= 8^2T(2^{l-2}) + \tfrac{1}{B} * 2^{2l+1} + \tfrac{1}{B} * 2^{2l} \\
&\quad \vdots \\
&= 8^{l-k}T(2^k) + \tfrac{1}{B} * 2^{3l-k-1} + ... + \tfrac{1}{B} * 2^{2l} \\
&= O((\tfrac{N}{\sqrt{M}})^3 * \tfrac{M}{B}) \\
&= O(N^3/B\sqrt{M})
\end{aligned}$$

Problem 2

(1) $T(N) = 8T(N/2) + \frac{N^2}{B}$

$T(N) = \frac{N^2}{B}$ when $N \leq \sqrt{M}$

(2) Let $N = 2^l$ and $2^k = \sqrt{M}$, thus $2^{l-k} = \frac{N}{\sqrt{M}}$

$$\begin{aligned}
T(2^l) &= 8T(2^{l-1}) + \frac{1}{B} * 2^{2l} \\
&= 8(8T(2^{l-2}) + \frac{1}{B} * 2^{2(l-1)}) + \frac{1}{B} * 2^{2l} \\
&= 8^2 T(2^{l-2}) + \frac{1}{B} * 2^{2l+1} + \frac{1}{B} * 2^{2l} \\
&\quad \vdots \\
&= 8^{l-k} T(2^k) + \frac{1}{B} * 2^{3l-k-1} + ... + \frac{1}{B} * 2^{2l} \\
&= O((\frac{N}{\sqrt{M}})^3 * \frac{M}{B}) \\
&= O(N^3/B\sqrt{M})
\end{aligned}$$

Problem 3

(1) $T(N) = 2T(N/2) + \frac{N}{B}$

$T(N) = \frac{N}{B}$ when $N \leq M$

(2) Let $N = 2^l$ and $2^k = M$, thus $2^{l-k} = \frac{N}{M}$

$$\begin{aligned}
T(2^l) &= 2T(2^{l-1}) + \frac{1}{B} * 2^l \\
&= 2(2T(2^{l-2}) + \frac{1}{B} * 2^{l-1}) + \frac{1}{B} * 2^l \\
&= 2^2 T(2^{l-2}) + \frac{1}{B} * 2^l + \frac{1}{B} * 2^l \\
&\quad \vdots \\
&= 2^{l-k} T(2^k) + \frac{1}{B} * 2^l + ... + \frac{1}{B} * 2^l \\
&= \frac{N}{M} * \frac{M}{B} + (l-k) * \frac{2^l}{B} \\
&= \frac{N}{B} + \frac{N}{B} * \lg \frac{N}{M} \\
&= O(\frac{N}{B} \log \frac{N}{M})
\end{aligned}$$

Problem 4

(1) First, we check $a[0], a[1], a[2], a[4], a[8]...a[2^k]$, until $a[2^k]$ is 1. Then we do binary search between $a[2^{k-1}]$ and $a[2^k]$. The time complexity is $O(logn)$

(2) First, we check $a[0], a[1], a[2], a[4], a[16]...a[2^{2^k}]$, until $a[2^{2^k}]$ is 1. Then we do this kind of search between $a[2^{2^{k-1}}]$ and $a[2^{2^k}]$. The time complexity is $O(loglogn)$

(3) First, we check $a[0], a[1], a[2], a[4], a[256], a[256^256]...a[k], a[k^k]$, until $a[k^k]$ is 1. Then we do this kind of search between $a[k]$ and $a[k^k]$. This is faster than (2).

Problem 5

Assign one prisoner to be the "counter". He will count how many different prisoners have been put into the room.

The rules are:

1. If you are not the counter:

    If the light is OFF, and you have never turned the light ON before, turn it ON.

    If the light is ON, do nothing.

2. If you are the counter:

    If the light is OFF, do nothing.

    If the light is ON, turn it OFF, and increase the counter by 1.

    If counter = n, announce that all prisoners have visited.

Reference: `http://www.ocf.berkeley.edu/~wwu/papers/100prisonersLightBulb.pdf`